Meta-PBS: Compact High-Precision Programmable Bootstrapping

Shihe Ma
msh24@mails.tsinghua.edu.cn
Institute for Network Sciences and Cyberspace, Tsinghua University
Beijing, China

Tairong Huang
huangtr@mail.tsinghua.edu.cn
Institute for Advanced Study, Tsinghua University
Beijing, China

Anyu Wang*†‡§
anyuwang@tsinghua.edu.cn
Institute for Advanced Study, Tsinghua University
Beijing, China

Changtong Xu
xuchangtong.xct@antgroup.com
Ant Group
Hangzhou, China

Tao Wei
lenx.wei@antgroup.com
Ant Group
Hangzhou, China

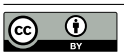
Xiaoyun Wang¶‖**
xiaoyunwang@tsinghua.edu.cn
Institute for Advanced Study, Tsinghua University
Beijing, China

Abstract

Programmable bootstrapping (PBS) in TFHE is a powerful primitive for homomorphic lookup-table (LUT) evaluation, but its practical precision is limited to $O(\sqrt{N})$ for RLWE dimension N. Existing solutions either enlarge the RLWE dimension or digit-decompose plaintexts before encryption, thus degrading either bootstrapping efficiency or the efficiency of downstream linear combinations.

We propose *Meta-PBS*, a single-ciphertext high-precision PBS framework that supports plaintext modulus up to $2N$ without increasing the RLWE dimension. The key idea is to view PBS through a redundancy-perturbation perspective: Meta-PBS starts from an under-redundant LUT and iteratively increases its redundancy by combining blind rotations with LWE-to-RLWE packing. Under this view, classical TFHE PBS becomes the zero-iteration special case of Meta-PBS. Asymptotically, Meta-PBS supports PBS with a plaintext modulus $2N$ with only $O(1)$ blind rotations, leading to an $O(\sqrt{N})$ speedup compared with PBS. Another key advantage of Meta-PBS is its additive noise growth, which enables more efficient parameters and stronger support for post-bootstrap homomorphic linear combinations. In practice, Meta-PBS evaluates a 12-bit negacyclic LUT in 166 ms, achieving an 81.0× speedup over EBS. For an arbitrary 8-bit LUT, a high-capacity Meta-WoP-PBS configuration is 2.04× faster than Refined LHE while supporting post-bootstrap linear combinations of size up to 25,000; a lower-capacity configuration reaches a 2.61× speedup.

*Corresponding author.
†Also with State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University.
‡Also with Zhongguancun Laboratory.
§Also with National Financial Cryptography Research Center.
¶Also with State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University.
‖Also with School of Cryptologic Science and Engineering, Shandong University.
**Also with Shandong Institute of Blockchain.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CCS '26, The Hague, Netherlands
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846610

CCS Concepts

• **Security and privacy → Public key encryption**.

Keywords

Fully Homomorphic Encryption; TFHE Scheme; Programmable Bootstrapping; High Precision

ACM Reference Format:
Shihe Ma, Tairong Huang, Anyu Wang, Changtong Xu, Tao Wei, and Xiaoyun Wang. 2026. Meta-PBS: Compact High-Precision Programmable Bootstrapping. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3830454.3846610

1 Introduction

Fully Homomorphic Encryption (FHE) enables arbitrary computation over encrypted data [23]. Most practical FHE schemes are built from noisy encryption such as LWE [22] and Ring-LWE (RLWE) [19]. As homomorphic evaluation proceeds, ciphertext noise grows and eventually causes decryption failure. Gentry's bootstrapping technique [12], which refreshes a ciphertext by resetting its noise level, is therefore the key ingredient that turns leveled homomorphic encryption into fully homomorphic encryption.

Among current bootstrapping techniques, the bootstrapping procedure in FHEW/TFHE [8, 11] is particularly attractive due to its low latency and its ability to evaluate a lookup table (LUT) during bootstrapping at essentially no additional cost. This makes TFHE bootstrapping a powerful primitive for general function evaluation, and explains why it is commonly referred to as programmable bootstrapping (PBS) [9] or functional bootstrapping (FBS) [5]. However, TFHE PBS also suffers from a fundamental precision bottleneck. Since the input message is encoded in the exponent of X over $\mathbb{Z}[X]/(X^N+1)$, the effective input modulus is bounded by $2N$. At the same time, after modulus switching, the perturbation on that exponent is dominated by a noise term of magnitude $O(\sqrt{N})$ for uniform binary secrets. As a result, the usable plaintext modulus is typically limited to $\frac{2N}{O(\sqrt{N})} = O(\sqrt{N})$, which is far below the maximum exponent range supported by the ring. This limited precision has become a major obstacle to deploying PBS in practical high-precision settings.

Existing approaches to overcoming this limitation mainly follow two directions. The first enlarges the RLWE dimension in order to support a larger plaintext modulus. A representative example is Extended Bootstrapping (EBS), which increases the ring dimension and uses a subring secret to control noise growth and improve parallelism [16]. The second direction represents a high-precision message by multiple ciphertexts. For instance, the leveled homomorphic evaluation (LHE) mode of TFHE encrypts plaintext data bitwise, uses external products for binary arithmetic, and relies on circuit bootstrapping (CBS) for noise management [8, 26]. Similarly, the tree-based/chaining approach decomposes a high-precision message into multiple ciphertexts and combines multiple bootstrappings with LWE-to-RLWE packings to evaluate large-precision LUTs [13, 24]. Refined LHE [25] also starts from digit-decomposed inputs, and further converts the message into RGSW ciphertexts encrypting plaintext bits so that an arbitrary LUT can be evaluated via external products with a plaintext polynomial.

These two directions leave an important design point only weakly addressed: *high-precision PBS on a single LWE ciphertext, without enlarging the RLWE dimension or requiring digit-decomposed inputs.* This design point matters for at least two reasons. First, the output of upstream homomorphic computation can be a standard, non-decomposed LWE ciphertext rather than a bit- or digit-decomposed representation. Second, downstream homomorphic linear combinations are particularly cheap in the single-ciphertext setting, since they can be implemented by ciphertext addition and constant multiplication. In contrast, when the message is split across many ciphertexts, even a linear combination may require multiple bootstrappings or much heavier auxiliary operations.

1.1 Our Contributions

Our contributions are as follows.

- **Single-ciphertext high-precision PBS.** We propose Meta-PBS, a generalization of TFHE PBS that evaluates negacyclic LUTs with plaintext modulus up to $2N$ without increasing the RLWE dimension. Its key idea is to view PBS through a redundancy-perturbation perspective: Meta-PBS iteratively increases the encoding redundancy of the blind-rotated LUT while keeping the exponent perturbation close to the modulus-switching scale. Under this view, classical TFHE PBS becomes the zero-iteration special case of Meta-PBS.
  Asymptotically, Meta-PBS evaluates a size-$2N$ negacyclic LUT using only $O(1)$ blind rotations in dimension N. This avoids the ring-dimension expansion to $O(N^{1.5})$ incurred by PBS and EBS, yielding an $O(\sqrt{N})$ speedup.
  Meta-PBS also has additive noise growth: noise contributions from blind rotations and RLWE packing are summed in the output noise, enabling efficient parameters and more post-bootstrap computation.
- **High-precision advanced PBS variants.** We further instantiate the Meta-PBS framework in the PBSmanyLUT and WoP-PBS settings [10], obtaining *Meta-PBSmanyLUT* for high-precision multi-output LUT evaluation and *Meta-WoP-PBS* for high-precision arbitrary non-negacyclic LUT evaluation. This carries the Meta-PBS advantage beyond the basic negacyclic single-output setting.
- **Concrete performance gains.** Meta-PBS evaluates a 12-bit negacyclic LUT in 166 ms, achieving an 81.0× speedup over EBS [16]. In the comparable core single-ciphertext LUT-evaluation phase, it is 21.1× faster than the operation-level estimate for LargeLUT [21]. For an arbitrary 8-bit LUT, the high-capacity Meta_8 configuration is 2.04× faster than the single-input Refined LHE configuration [25] while supporting $\mathbb{Z}$-linear combinations of more than 25,000 ciphertexts, over 25 times as many as Refined LHE. The lower-capacity Meta_8C configuration reaches a 2.61× speedup.

1.2 Technical Overview

We begin with a high-level reinterpretation of classical TFHE PBS. Let $c = (\vec{a}, b) \in \mathbb{Z}_q^{n+1}$ be an LWE ciphertext encrypting $m \in \mathbb{Z}_t$. After modulus switching to $2N$, its linear decryption has the form $\frac{2N}{t}m + e'$, where the exponent perturbation e' is dominated by the modulus-switching error. To evaluate a negacyclic LUT, TFHE PBS blind-rotates a test vector whose coefficients around the correct LUT entry are repeated over a contiguous window. As long as the perturbation e' is smaller than roughly half of that window, the constant term of the blind-rotation result is unaffected. In other words, *classical PBS succeeds because the clean test vector has enough redundancy to absorb the perturbation*; see Figure 1.

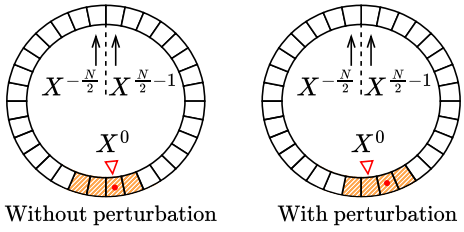


Figure 1: The blind-rotated test vector in TFHE PBS, with and without perturbation on the exponent. The redundancy absorbs the perturbation to ensure a correct result.

Meta-PBS keeps this basic principle but removes the requirement that the initial test vector must already have enough redundancy. The first step of Meta-PBS is still performing the initial blind rotation, but the correct LUT entry cannot be extracted from this result due to the insufficient redundancy. Yet, Meta-PBS will increase the desired entry's redundancy by alternately performing the following two steps.

Step 1: amplify redundancy. At this point, the desired LUT entry is located near the constant term, but it cannot be extracted correctly due to insufficient redundancy. Meta-PBS applies a LWE-to-RLWE packing map called *HomTruncRepeat*, which keeps only a small neighborhood around the constant term and repeats each surviving coefficient several times. This increases the local redundancy around the desired value. For the framework description,

HomTruncRepeat can be treated as a black-box LWE-to-RLWE packing map. Its optimized construction is delayed to Section 5.

Step 2: refine the perturbation. The perturbation introduced by modulus switching is not arbitrary: it comes from lower bits discarded during the modulus-switching process. Meta-PBS extracts these lower bits approximately through a quotient-and-remainder style decomposition of the input ciphertext. This produces a sequence of LWE ciphertexts $c_0, c_1, \ldots$, where c_0 drives the first blind rotation and later ciphertexts encode the modulus-switching noise in the previous ciphertext. A subsequent blind rotation then transfers the amplified redundancy to a new test vector while replacing the old perturbation by a new one that remains at essentially the modulus-switching scale.

By iterating these two steps, Meta-PBS gradually pushes the redundancy-perturbation ratio beyond the threshold needed for correctness. This enables bootstrapping to start from a non-redundant LUT, including the extreme case $t = 2N$. Under this interpretation, standard TFHE PBS is simply the special case in which the initial redundancy is already sufficient and no iteration is needed.

An example with $t = 2N$ and one iteration is illustrated in Figure 2. On the right, the ciphertexts used in successive blind rotations are derived from the input ciphertext through quotient-and-remainder extraction. The left side tracks the evolution of the blind-rotated test vector: the first blind rotation places the correct entry near the constant term, HomTruncRepeat amplifies both its redundancy and perturbation by β, and the second blind rotation reduces the remaining perturbation to a smaller offset that can be safely absorbed.

This iterative refinement has a similar high-level flavor to META-BTS for CKKS bootstrapping [2], although the concrete mechanism in Meta-PBS is specific to TFHE-style exponent encoding and blind rotation.

1.3 Related Works

Our work is most closely related to recent efforts to overcome the precision bottleneck of TFHE-style LUT evaluation. As discussed above, existing approaches mainly either enlarge the ring dimension or decompose a high-precision message across multiple ciphertexts. Meta-PBS differs from both by targeting high-precision LUT evaluation directly on a single LWE ciphertext, without ring expansion.

A particularly relevant parallel work is LargeLUT by Nishimura et al. [21] (CCS'25), which also achieves high-precision LUT evaluation through iterative blind rotations. The two works start from different input models: LargeLUT assumes a digit-decomposed message distributed across multiple LWE ciphertexts, whereas Meta-PBS takes a single standard LWE ciphertext as input. Therefore, the two works are not directly comparable as end-to-end solutions. However, after preprocessing, LargeLUT also reduces its task to a high-precision, single-ciphertext LUT evaluation phase. At the level of this *core single-ciphertext phase*, the two approaches can be meaningfully compared.

In this core phase, Meta-PBS exhibits lower complexity for the following reasons. First, after preprocessing, LargeLUT still requires $O(\log t)$ blind rotations, whereas Meta-PBS needs only $O(1)$. Second, Meta-PBS uses a more specialized RLWE-packing procedure, *HomTruncRepeat**, in place of the regular PackingKS used by LargeLUT, leading to lower running time and smaller noise growth. Third, LargeLUT requires the repetition base β to be a power of two, whereas Meta-PBS removes this restriction. This extra flexibility can reduce the number of blind rotations under some parameter choices.

1.4 Scope and discussion.

Meta-PBS is designed for univariate LUT evaluation on a *single*, non-decomposed LWE ciphertext with $t \le 2N$, especially when the bootstrapped output must support a large number of additions or constant multiplications. This scenario is where Meta-PBS is more suitable than decomposition-based methods, such as Refined LHE and tree-based methods. In contrast, decomposition-based approaches are preferable in three regimes. First, if the inputs are already bit- or digit-decomposed, as in multivariate LUT evaluation, they can operate directly on that representation, whereas Meta-PBS would require expensive preprocessing to combine the inputs into one ciphertext. Second, for $t > 2N$, they avoid the exponent-range limit of a single ciphertext. Third, bitwise and digit-wise functions often decompose into a few low-precision blind rotations and therefore admit faster homomorphic evaluation in a decomposed representation.

Meta-PBS should therefore be understood not as a replacement for all high-precision LUT-evaluation methods, but as a complementary design point that substantially advances the efficiency frontier of single-ciphertext PBS. Its value, however, is not limited to serving as a faster PBS method in this setting. Single-ciphertext PBS is also a basic building block in a number of more complex high-precision constructions, including decomposition-based PBS variants and homomorphic digit-decomposition procedures [18]. Therefore, even in regimes where Meta-PBS itself is not the most suitable end-to-end solution, it may still improve those methods by replacing their small-precision single-ciphertext PBS components with an efficient high-precision alternative.

2 Preliminaries

2.1 Notations

For an integer $q \ge 2$, define $[q] = \mathbb{Z} \cap [0, q-1]$ and $[q]_{\mathrm{sym}}$ as $\mathbb{Z} \cap [-\lfloor \frac{q}{2} \rfloor, \lceil \frac{q}{2} \rceil - 1]$, with $[2]_{\mathrm{sym}} = \{0, 1\}$ as the only exception. We identify the representatives of $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ with $[q]_{\mathrm{sym}}$. $[a]_q^{\mathrm{sym}}$ and $[a]_q$ return the unique elements in $[q]_{\mathrm{sym}}$ and $[q]$, respectively, that are congruent to a modulo q. $[\cdot]_q^{\mathrm{sym}}, [\cdot]_q$ are applied to integer vectors and integer sets in the natural way. We use $[\![a, b]\!]$ to denote $\mathbb{Z} \cap [a, b]$.

For a polynomial $f(X)$, the coefficient corresponding to X^i is denoted by $f(X)[i]$. The support of a polynomial is defined as $\mathrm{supp}(f(X)) = \{i \mid f(X)[i] \neq 0\}$. With N a power of two, the $2N$-th cyclotomic ring is $R = \mathbb{Z}[X]/(X^N + 1)$. For an integer $q \ge 2$, $R_q = R/qR$. Elements in R are represented by polynomials of degree $< N$. For $A \in R$, its coefficient vector is denoted as $\vec{A} = (A[0], A[1], \cdots, A[N-1])$.

Gadget decomposition $G^{(B,l,q)}$ is parametrized by a base B, a level l, and a modulus q. We assume $B^l | q$. For $a \in \mathbb{Z}_q$, let $\vec{g}^{(B,l,q)} =$

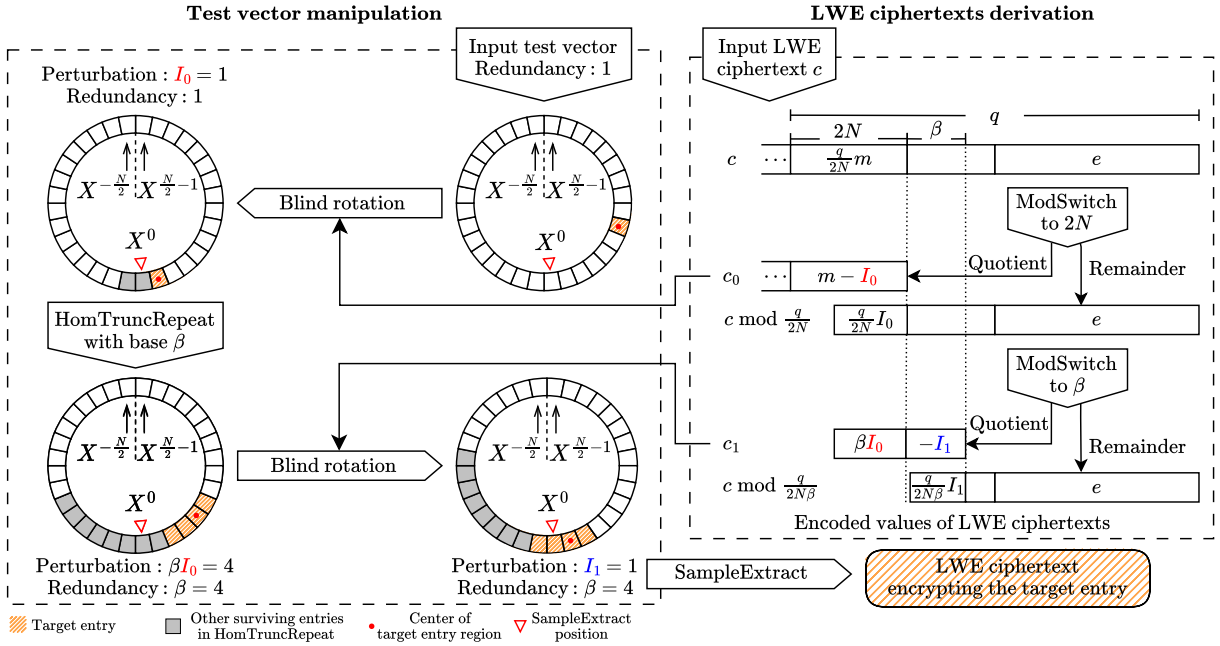


Figure 2: The workflow of Meta-PBS with $t = 2N$ and one iteration.

$(\frac{q}{B^l}, \cdots, \frac{q}{B^2}, \frac{q}{B}) \in \mathbb{Z}^l$, $G^{(B,l,q)}(a) \in \mathbb{Z}^l$ satisfies $G^{(B,l,q)}(a) \in [B]^l_{\text{sym}}$ and $a - \langle G^{(B,l,q)}(a), \vec{g}^{(B,l,q)} \rangle \in [\frac{q}{B^l}]_{\text{sym}}$. $G^{(B,l,q)}(\cdot)$ is applied to integer vectors in the natural way. We omit q in the superscript when it is clear from the context.

For a finite set S, $a \leftarrow S$ means a is uniformly sampled from S. For a distribution χ, $a \leftarrow \chi$ means a is sampled according to χ. The variance of a random variable a is denoted as $\text{Var}(a)$.

2.2 LWE and RLWE Encryption

LWE encryption of a plaintext $m \in \mathbb{Z}$ produces a ciphertext $c = (\vec{a}, b = \langle \vec{a}, \vec{s} \rangle + m + e) \in \mathbb{Z}_q^{n+1}$, where $\vec{a}_i \leftarrow \mathbb{Z}_q$, $s \in \mathbb{Z}^n$ is the secret key, e is the encryption noise, and n, q are the LWE dimension and ciphertext modulus. We instantiate LWE with dense binary secrets and Gaussian noise, i.e., $\vec{s}_i \leftarrow \{0,1\}$, $e \leftarrow N(0, \sigma^2_{\text{LWE}})$. $\text{LWE}_{\vec{s},q}(m; e)$ represents an LWE ciphertext modulo q with secret $\vec{s}$, message m and noise e, where $\vec{s}, q, e$ are sometimes omitted. For $c = (\vec{a}, b) = \text{LWE}_{\vec{s},q}(m; e)$, its phase is defined as $\psi(c) = [b - \langle \vec{a}, \vec{s} \rangle]_q = [m + e]^{\text{sym}}_q$, and its raw phase is defined as $\psi^*(c) = b - \langle \vec{a}, \vec{s} \rangle \in \mathbb{Z}$, which is $\psi(c)$ without modular reduction.

A Lev ciphertext encrypting m is a list of LWE ciphertexts encrypting the entries of $m \cdot \vec{g}^{(B,l,q)}$, i.e., $\text{Lev}^{(B,l)}_{\vec{s},q}(m) = (\text{LWE}_{\vec{s},q}(m \cdot \frac{q}{B^l}), \cdots, \text{LWE}_{\vec{s},q}(m \cdot \frac{q}{B}))$. The gadget product of $m \in \mathbb{Z}_q$ and $\text{Lev}^{(B,l)}_{\vec{s},q}(v)$ is denoted as $m \odot \text{Lev}^{(B,l)}_{\vec{s},q}(v) = \langle G(m), \text{Lev}^{(B,l)}_{\vec{s},q}(v) \rangle$. The superscripts and subscripts in LWE and Lev notations are sometimes omitted for clarity.

RLWE encryption of a plaintext $m \in R$ produces $C = (A, B = AS + E) \in R^2_Q$, with $S \in R$ as the secret, E as the noise, and Q as the ciphertext modulus. Still, $S[i] \leftarrow \{0,1\}$ and $E[i] \leftarrow N(0, \sigma^2_{\text{RLWE}})$. $\psi(\cdot)$, $\text{RLWE}(\cdot)$, and $\text{RLev}(\cdot)$ are defined similarly to the LWE case.

2.3 TFHE Overview

The TFHE scheme uses RLWE to bootstrap LWE ciphertexts and simultaneously evaluate a LUT on the LWE plaintext. Let $\vec{s} \in \mathbb{Z}^n$ be the LWE secret, and $S \in R$ be the RLWE secret. LWE and RLWE have the same ciphertext modulus q. TFHE bootstrapping consists of the following building blocks.

- **InitTestVector**$(f : \mathbb{Z}_t \to \mathbb{Z}_t)$:
  Output $\text{TV} = X^{-N/t} \sum_{i=0}^{t/2-1} \left(\frac{q}{t} f(i) X^{2Ni/t} \cdot \sum_{j=0}^{2N/t-1} X^j \right)$.
- **ModSwitch**$(\text{LWE}_{\vec{s},q}(\frac{q}{t}m; e) = (\vec{a}, b), q')$:
  Output $\text{LWE}_{\vec{s},q'}(\frac{q'}{t}m; \frac{q'}{q}e + e_{\text{ms}}) = (\lfloor \frac{q'}{q}\vec{a} \rceil, \lfloor \frac{q'}{q}b \rceil)$ with e_{ms} as the modulus-switching noise.
- **BlindRotate**[1]$(c = \text{LWE}_{\vec{s},q}(m'; e), \text{RLWE}_S(M; E), \text{BRK}^{(B_{\text{br}},l_{\text{br}})})$:
  Output $\text{RLWE}_S(M \cdot X^{-\psi^*(c)}); E + E_{\text{br}})$ with E_{br} as the blind rotation noise. $\text{BRK}^{(B_{\text{br}},l_{\text{br}})}$ is the blind rotation key.
- **SampleExtract**$(C = (A, B) = \text{RLWE}_S(M; E), i \in [N])$:
  Output $c = \text{LWE}_{\vec{S},q}(M[i]; E[i])$.
- **KeySwitch**$(\text{LWE}_{\vec{s},q}(m'; e), \text{KSK}^{(B_{\text{ks}},l_{\text{ks}})}_{\vec{s}\to\vec{s}'})$:
  Output $\text{LWE}_{\vec{s}',q}(m'; e + e_{\text{ks}})$, with e_{ks} as the key-switching noise. $\text{KSK}^{(B_{\text{ks}},l_{\text{ks}})}_{\vec{s}\to\vec{s}'}$ is the key-switching key.

On input $c = \text{LWE}_{\vec{s},q}(\frac{q}{t}m; e)$ and a negacyclic function $f(x) : \mathbb{Z}_t \to \mathbb{Z}_t$, TFHE PBS computes

$$c' \leftarrow \text{ModSwitch}(c, 2N),$$
$$\text{TV} \leftarrow \text{InitTestVector}(f),$$
$$c_{\text{br}} \leftarrow \text{SampleExtract}(\text{BlindRotate}(c', \text{TV}, \text{BRK}^{(B_{\text{br}},l_{\text{br}})}), 0).$$

[1] BlindRotate can operate on both RLWE ciphertexts and plaintext polynomials.

Homomorphic computation, like additions or scalar multiplications, is performed on c_{br} and other LWE ciphertexts, resulting in another LWE ciphertext c_{out}. This ciphertext is then key-switched back to $\vec{s}$ before being fed into the next PBS

$$c_{\text{next}} \leftarrow \text{KeySwitch}(c_{\text{out}}, \text{KSK}_{\vec{S}\to\vec{s}}^{(B_{\text{ks}},l_{\text{ks}})})$$

Our noise analysis is based on the variance estimates from Chillotti et al. [10] for e_{ms}, E_{br}, and e_{ks}, while the FFT noise in E_{br} is estimated using the result by Bossuat et al. [3]. e_{ks} is reduced using the shared randomness secret keys [4, 15].

$$\begin{aligned} V_{\text{br}} := \text{Var}(E_{\text{br}}) &= 2nNl_{\text{br}}\frac{B_{\text{br}}^2+2}{12}\sigma_{\text{BRK}}^2 + n(\frac{N}{2}+1)\frac{q^2-B_{\text{br}}^{2l_{\text{br}}}}{24B_{\text{br}}^{2l_{\text{br}}}} \\ &\quad + \frac{nN}{32} + \frac{n}{16}(\frac{N}{2}-1)^2 + 2^{19.4}\cdot 2nN^2 l_{\text{br}}B_{\text{br}}^2, \\ V_{\text{ks}} := \text{Var}(e_{\text{ks}}) &= (N-n)\frac{q^2-B_{\text{ks}}^{2l_{\text{ks}}}}{24B_{\text{ks}}^{2l_{\text{ks}}}} + \frac{N-n}{16} \\ &\quad + (N-n)l_{\text{ks}}\sigma_{\text{ksk}}^2\frac{B_{\text{ks}}^2+2}{12}, \end{aligned}$$

and the distribution of e_{ms} is given in Lemma 3.

2.4 LWE-to-RLWE Packing

There are various methods to pack multiple LWE ciphertexts into an RLWE ciphertext, whose polynomial coefficients consist of the input plaintexts.

Given z LWE ciphertexts $\{(\vec{a}^{(i)}, b^{(i)}) = \text{LWE}_{\vec{s}}(m^{(i)}; e^{(i)})\}_{i\in[z]}$, denote the LWE-to-RLWE packing function as a $\mathbb{Z}$-linear function $f : \mathbb{Z}_q^z \to R_q$. The public functional key-switching [8] can homomorphically evaluate f.

$$\begin{aligned} &\underline{\textbf{PubFuncKS}}(f, \{(\vec{a}^{(i)}, b^{(i)})\}_{i\in[z]}, \{\text{RLev}_S^{(B,l)}(\vec{s}_j)\}_{j\in[n]}) \\ &= \sum_{j\in[n]} f(\{\vec{a}_j^{(i)}\}_{i\in[z]}) \odot \text{RLev}_S^{(B,l)}(\vec{s}_j) + f(\{b^{(i)}\}_{i\in[z]}) \\ &= \text{RLWE}_S(f(\{m^{(i)}\}_{i\in[z]}); f(\{e^{(i)}\}_{i\in[z]}) + E_{\text{pfks}}), \end{aligned}$$

where E_{pfks} is the noise added by packing.

Suppose each input LWE plaintext is packed into B consecutive RLWE coefficients, and let $P_B = \sum_{k\in[B]} X^k$. The **base-aware PackingKS** [13] moves P_B from $f(\{m^{(i)}\}) = \sum_{i\in[z]} m^{(i)}X^{iB}P_B$ into the RLev ciphertexts, reducing the variance of E_{pfks} by a factor of B. Also, when z is small, each polynomial-polynomial product between $G^{(B,l,q)}(f(\{\vec{a}_j^{(i)}\}_{i\in[z]}))$ and $\text{RLev}_S^{(B,l)}(\vec{s}_j \cdot \sum_{k\in[B]} X^k)$ can be replaced by z integer-polynomial products, reducing the computational cost.

In the special case where the input LWE ciphertexts for LWE-to-RLWE packing are extracted from the same RLWE ciphertext, the packing function $f : \mathbb{Z}_q^z \to R_q$ can be expressed as a function $F : R_q \to R_q$. For example, $F(M) = \sum_{i\in[z]} M[i]\cdot X^{iB}\cdot\sum_{k\in[B]} X^k$ packs and duplicates the first z coefficients in an RLWE ciphertext. The public functional key-switching now has the form of

$$\begin{aligned} &\underline{\textbf{PackRLWE}}(F, \text{RLWE}_S(M;E) = (A,B), \{\text{RLev}_S^{(B,l)}(S[i])\}_{i\in[N]}) \\ &= \sum_{i\in[N]} F(A\cdot X^i) \odot \text{RLev}_S^{(B,l)}(S[i]) + F(B) \\ &= \text{RLWE}_S(F(M); F(E) + E_{\text{pfks}}). \end{aligned}$$

The base-aware PackingKS technique still applies to this case.

3 The Meta-PBS Framework

This section formally describes the workflow of Meta-PBS to evaluate high-precision negacyclic functions homomorphically. The conditions for correctness, output noise size, and asymptotic complexity will also be analyzed.

3.1 The Redundancy-Perturbation Perspective

As suggested by the toy example in Section 1.2, the key idea of Meta-PBS is to progressively increase the encoding redundancy of the blind-rotated test vectors while keeping the perturbation on the exponent small. This idea can be formalized through a redundancy-perturbation framework, which, notably, also offers a unified way to interpret the classical TFHE PBS. Accordingly, we first revisit TFHE PBS from this perspective, and then show how Meta-PBS fits into the same framework and how it differs from TFHE PBS.

Recall that in TFHE PBS, the input LWE ciphertext $c \in \mathbb{Z}_q^{n+1}$ is modulus-switched to $2N$, producing $c_0 = \text{LWE}(\frac{2N}{t}m; e_0) \in \mathbb{Z}_{2N}^{n+1}$. For a negacyclic function $f : \mathbb{Z}_t \to \mathbb{Z}_t$ (i.e., $f(x+t/2) = -f(x)$), the blind rotation homomorphically multiplies the test vector

$$\text{TV} = X^{-N/t}\sum_{i=0}^{t/2-1}\left(\frac{q}{t}f(i)X^{2Ni/t}\cdot\sum_{j=0}^{2N/t-1}X^j\right) \in R_q$$

by $X^{-\psi^*(c_0)} = X^{-2N/t\cdot m - e_0}$.

The blind rotation result $\text{TV}\cdot X^{-2N/t\cdot m-e_0}$ is the product of two parts, namely the perturbation-free polynomial $\text{TV}_{\text{clean}} = \text{TV}\cdot X^{-2N/t\cdot m}$, and the perturbation term X^{-e_0}. Thanks to the redundancy in TV_{clean}, the constant coefficient of $\text{TV}_{\text{clean}}\cdot X^{-e_0}$ is equal to $\frac{q}{t}f(m)$ as long as $e_0 \in [\![-N/t, N/t-1]\!]$. TFHE PBS exploits this property to remove the effect of e_0 on the bootstrapping result, resulting in an encryption of $f(m)$ as long as the perturbation $|e_0|$ is small enough, specifically $e_0 \in [2N/t]_{\text{sym}}$. We translate this correctness condition for PBS into the following concept of *r-redundancy*.

Definition 1. *$M \in R_q$ is called r-redundant in $v \in \mathbb{Z}_q$ if*

$$\forall i \in [r]_{\text{sym}}, (M\cdot X^{-i})[0] = v.$$

Using the language of Definition 1, in TFHE PBS, $\text{TV}\cdot X^{-2N/t\cdot m}$ is $2N/t$-redundant in $\frac{q}{t}f(m)$, and the PBS outputs the correct result if $2|e_0|+1 \le 2N/t$.

This redundancy requirement of TFHE PBS limits the choice of t to $t \le \frac{2N}{2|e_0|+1}$. Since $|e_0|$ is dominated by the modulus-switching noise and has an upper bound of $O(\sqrt{N})$, the classical TFHE PBS can only achieve $t = O(\sqrt{N})$.

3.2 Iteratively Increasing the Redundancy

In contrast to TFHE PBS, Meta-PBS does not require the initial TV to have enough redundancy to remove the perturbation. Instead, it starts with an under-redundant TV and increases its redundancy through iterative blind rotations. This allows it to support t as large as $2N$. The Meta-PBS workflow is summarized as follows.

Initial blind rotation. We compute c_0 and blind-rotate TV to obtain $\mathrm{RLWE}(M_0)$, where

$$c_0 = \mathrm{ModSwitch}(c, 2N) = \mathrm{LWE}(\tfrac{2N}{t}m; e_0),$$
$$M_0 = \mathrm{TV} \cdot X^{-2N/t \cdot m - e_0}.$$

Recall that

$$M_0 \cdot X^{e_0} \text{ is } \frac{2N}{t}\text{-redundant in } \frac{q}{t}f(m). \tag{1}$$

This first step is identical to TFHE PBS, except that M_0 is not redundant enough to prevent X^{e_0} from affecting its constant term.

Increasing redundancy through HomTruncRepeat. In the i-th iteration ($i = 0, 1, \cdots$), let e_i denote the exponent perturbation in M_i, so that $M_i \cdot X^{e_i}$ is the corresponding perturbation-free test vector. Meta-PBS first uses LWE-to-RLWE packing with base β_i to produce $\mathrm{RLWE}(M_i')$, where

$$M_i' \cdot X^{\beta_i e_i} \text{ is (at most) } \prod_{j=0}^{i} \beta_j \frac{2N}{t}\text{-redundant in } \frac{q}{t}f(m). \tag{2}$$

That is, the i-th iteration increases both the redundancy and perturbation by a factor of β_i.

Generally speaking, the packing process extracts the coefficients that are near the constant term of M_i and repeats each one β_i times. Thus, we call this packing algorithm HomTruncRepeat, and its plaintext version TruncRepeat.

Decreasing the perturbation. The second part of the i-th iteration consists of a blind rotation. The LWE ciphertext used in this blind rotation, denoted by $c_{i+1} \in \mathbb{Z}_{\beta_i}^{n+1}$, can be derived from c and satisfies $\psi^*(c_{i+1}) + \beta_i e_i = e_{i+1}$. Denote the result of blind-rotating $\mathrm{RLWE}(M_i')$ by c_{i+1} as $\mathrm{RLWE}(M_{i+1})$. According to Equation 2,

$$M_{i+1} \cdot X^{e_{i+1}} \text{ is (at most) } \prod_{j=0}^{i} \beta_j \frac{2N}{t}\text{-redundant in } \frac{q}{t}f(m). \tag{3}$$

Importantly, e_i is dominated by a modulus-switching noise[2], which means the size of the perturbation remains almost unchanged after each iteration.

Repeating the iterations. By repeatedly performing HomTruncRepeat and blind rotations for K times, the redundancy in M_K may eventually become large enough so that it can cancel the effect of the perturbation e_K, i.e., $\prod_{i=0}^{K-1} \beta_i \frac{2N}{t} \geq 2|e_K| + 1$ holds with high probability. Extracting the constant term of M_K produces $\mathrm{LWE}(\frac{q}{t}f(m))$, marking the end of Meta-PBS.

3.3 A Formal Description of Meta-PBS

We begin with the definition of HomTruncRepeat and the derivation of c_i, before presenting the whole Meta-PBS algorithm.

3.3.1 Behavior of TruncRepeat. At this stage, we treat HomTruncRepeat as a black-box packing primitive and only specify its plaintext behavior. This keeps the framework presentation independent of any particular instantiation; an optimized realization is deferred to Section 5.

[2]This is straightforward for $i = 0$ and holds for other i if the initial LWE error in c is small enough.

Definition 2. *For integers $r > 0, a \leq 0 \leq b, \beta > 0$ and $M \in R_q$, the output of* $\mathrm{TruncRepeat}_r(M, [\![a, b]\!], \beta)$ *is defined as*

$$\left(\sum_{i \in [\![0,b]\!]} M[i] \cdot X^{i\beta} + \sum_{i \in [\![a,-1]\!]} M[N+i] \cdot X^{N+i\beta}\right) \cdot \sum_{j \in [\beta]} X^j \cdot X^{\Delta_{r,\beta}} \in R_q,$$

where $\Delta_{r,\beta} = \min [r\beta]_{\mathrm{sym}} - \beta \min [r]_{\mathrm{sym}}$.

Ignoring the term $X^{\Delta_{r,\beta}}$ in $\mathrm{TruncRepeat}_r(M, [\![a, b]\!], \beta)$, the remaining part has a coefficient vector of

$$(\underbrace{M[0], \cdots}_{\beta \text{ copies}}, \underbrace{M[1], \cdots}_{\beta \text{ copies}}, \cdots, \underbrace{M[b], \cdots}_{\beta \text{ copies}}, 0, \cdots,$$
$$\cdots, 0, \underbrace{M[N+a], \cdots}_{\beta \text{ copies}}, \cdots, \underbrace{M[N-1], \cdots}_{\beta \text{ copies}}).$$

This property of TruncRepeat is captured by the following lemma.

Lemma 1. *Let $M' = \mathrm{TruncRepeat}_r(M, [\![a, b]\!], \beta)$, then*

$M' \cdot X^{-\Delta_{r,\beta}} \cdot X^{-i\beta}$ is β-redundant in $(M \cdot X^{-i})[0]$ for $i \in [\![a, b]\!]$.

When M (or a cyclic shift of M) is already r-redundant in some v, TruncRepeat amplifies the redundancy by a factor of β. The term $X^{\Delta_{r,\beta}}$ is designed to center the positions of the redundant coefficients in this case. This is described in the following theorem.

Theorem 1 (Amplifying Redundancy). *If $M \cdot X^e$ is r-redundant in v, $(b - a + 1)\beta \leq N$, and $[r]_{\mathrm{sym}} - e \subseteq [\![a, b]\!]$, then*

$$\mathrm{TruncRepeat}_r(M, [\![a, b]\!], \beta) \cdot X^{e\beta} \text{ is } r\beta\text{-redundant in } v.$$

Proof. Let $M' = \mathrm{TruncRepeat}_r(M, [\![a, b]\!], \beta)$. M' has $(b-a+1)\beta$ consecutive non-zero coefficients, and $(b - a + 1)\beta \leq N$ ensures that these coefficients do not overlap in the ring $\mathbb{Z}_q[X]/(X^N + 1)$.

The r-redundancy of $M \cdot X^e$ implies that $(M \cdot X^{-i})[0] = v$ for $i \in [r]_{\mathrm{sym}} - e$. The condition of $[r]_{\mathrm{sym}} - e \subseteq [\![a, b]\!]$ allows us to use Lemma 1, resulting in

$$(M' \cdot X^{-\Delta_{r,\beta}} \cdot X^{-i\beta - j})[0] = v \text{ for } i \in [r]_{\mathrm{sym}} - e \text{ and } j \in [\beta].$$

Thus, $(M' \cdot X^{-k})[0] = v$ for $k = \Delta_{r,\beta} + \beta i + j$. It then follows from $i \in [r]_{\mathrm{sym}} - e$, $j \in [\beta]$ and $\Delta_{r,\beta} = \min[r\beta]_{\mathrm{sym}} - \beta \min[r]_{\mathrm{sym}}$ that

$$(M' \cdot X^{-k})[0] = v \text{ for } k \in [r\beta]_{\mathrm{sym}} - e\beta.$$

This directly implies $M' \cdot X^{e\beta}$ being $r\beta$-redundant in v. □

3.3.2 Extracting the modulus-switching noise. We use homomorphic approximate Euclidean division, denoted as HomDivRem, to extract c_i from c in each iteration.

It can be understood as a modulus switching with remainder, where c_{quo} is the output of the regular modulus switching, and c_{rem} is the remainder ciphertext whose raw phase stores information about the modulus-switching noise in c_{quo}.

Definition 3. *For an LWE ciphertext $c = (\vec{a}, b) \in \mathbb{Z}^{n+1}$ and a target quotient modulus $q' \mid q$, define* $\mathrm{HomDivRem}(c, q') = (c_{\mathrm{quo}}, c_{\mathrm{rem}})$, *where*

$$c_{\mathrm{rem}} = ([\vec{a}]_{\frac{q}{q'}}^{\mathrm{sym}}, [b]_{\frac{q}{q'}}^{\mathrm{sym}}) \in \mathbb{Z}^{n+1}, \tag{4}$$

$$c_{\mathrm{quo}} = \frac{q'}{q}(c - c_{\mathrm{rem}}) \in \mathbb{Z}^{n+1}. \tag{5}$$

Remark. The original version of homomorphic Euclidean division [7] sets $c_{\text{quo}} = \text{ModSwitch}(c, q') \in \mathbb{Z}_{q'}^{n+1}$, while our definition differs from it by not reducing c_{quo} modulo q'. This modification helps to preserve the information of $\psi^*(c)$ as

$$\psi^*(c) = \frac{q}{q'}\psi^*(c_{\text{quo}}) + \psi^*(c_{\text{rem}}). \tag{6}$$

The LWE ciphertexts for the blind rotations are derived as

$$c_0, c_{\text{rem},0} = \text{HomDivRem}(c, 2N), \tag{7}$$

$$c_i, c_{\text{rem},i} = \text{HomDivRem}(c_{\text{rem},i-1}, \beta_{i-1}), i = 1, 2, \cdots, K, \tag{8}$$

where $c_{\text{rem},i}$ has a ciphertext modulus of

$$q_i' = \frac{q}{2N\prod_{j\in[i]}\beta_j}.$$

Now we discuss several properties of HomDivRem that will be used later. Firstly, we show that the remainder ciphertext is closely related to the modulus switching noise in the quotient ciphertext.

Lemma 2. *For $c = \text{LWE}_{\vec{s}}(\frac{q}{t}m; e) \in \mathbb{Z}_q^{n+1}$ and $q' \mid q$,*

$$\text{ModSwitch}(c, q') = \text{LWE}_{\vec{s}}\left(\frac{q'}{t}m; \frac{q'}{q}\left(e - \psi^*([c]_{q'}^{\text{sym}})\right)\right),$$

i.e., the modulus switching noise is $-\psi^*([c]_{q'}^{\text{sym}})$.

Proof. Let $c_{\text{quo}}, c_{\text{rem}} = \text{HomDivRem}(c, q')$. Then the conclusion follows directly from $\text{ModSwitch}(c, q') \equiv c_{\text{quo}} \bmod q'$ and $\psi^*(c) = \frac{q}{q'}\psi^*(c_{\text{quo}}) + \psi^*(c_{\text{rem}})$. □

Secondly, the modulus switching noise (also the ψ^* value of an LWE ciphertext) is usually modeled heuristically as a Gaussian distribution with a standard deviation of $O(q\sqrt{n})$. This will later be used to bound the size of the perturbations e_i.

Lemma 3. *[10] For $c = \text{LWE}_{\vec{s},Q}(m; e)$ with $\vec{s} \leftarrow \{0,1\}^n$,*

$$\text{Var}(\psi^*(c)/Q) = \frac{n+2}{24} + \frac{1}{Q^2}\cdot\left(\frac{n-4}{48} + \text{Var}([m+e]_Q^{\text{sym}})\right).$$

$$E(\psi^*(c)/Q) = \frac{n-2}{4Q} + \frac{1}{Q}\cdot E([m+e]_Q^{\text{sym}}).$$

Lemma 2 is applied to c_{rem} in HomDivRem with $Q = \frac{q}{q'}$. For practical values of Q, e.g., $Q > 2^{50}$, some terms in the formula are negligible, leading to the following simplified result.

Proposition 1. *For $c = \text{LWE}_{\vec{s},q}(\frac{q}{t}m; e)$ and $q' \mid q$ with $\frac{q}{q'}$ sufficiently large, the noise e' in $\text{LWE}_{\vec{s},q'}(\frac{q'}{t}m; e') = \text{ModSwitch}(c, q')$ can be treated as a Gaussian distribution with*

$$\text{Var}(e') \le \frac{n+4}{24} + \left(\frac{q'}{q}\right)^2\cdot\text{Var}(e), \quad |E(e')| \le \frac{1}{2}.$$

Parameters and variables used in Meta-PBS are summarized in Table 1. The full Meta-PBS algorithm is given in Algorithm 1. We assume that the LWE noise in c follows a centered Gaussian distribution with a variance of V_{in}.

Table 1: Parameters and variables used by Meta-PBS.

Var.	Description
K	The number of iterations.
i	The indices of M_i, e_i, and r_i range from 0 to K; iteration indices range from 0 to $K-1$.
c_i	c_0 is used for the initial blind rotation; c_{i+1} is used for the blind rotation in iteration i.
q_i'	The modulus of $c_{\text{rem},i}$, namely $q_i' = q/(2N\prod_{j\in[i]}\beta_j)$.
M_i	For $i < K$, the encrypted RLWE polynomial before iteration i; M_K is the final output polynomial.
e_i	The perturbation before iteration i.
r_i	The redundancy of $M_i \cdot X^{e_i}$ in $\frac{q}{t}f(m)$.
β_i	HomTruncRepeat's packing base in the i-th iteration.
δ_i	A high probability upper bound on $\lvert e_i\rvert$.
T_i	The i-th HomTruncRepeat truncates to $[\![-T_i, T_i]\!]$.
p_{fail}	The failure probability when bounding e_i, i.e., $\Pr[\lvert e_i\rvert > \delta_i] \le p_{\text{fail}}$.

Algorithm 1 Meta-PBS

Require: Meta-PBS parameters from Table 1
Require: A negacyclic function f over $\mathbb{Z}_t$
Require: LWE ciphertext $c = \text{LWE}_{\vec{s}}(\frac{q}{t}m; e)$ with $\text{Var}(e) \le V_{\text{in}}$
Require: Blind rotation keys $\text{BRK}^{(B_{\text{br}}, l_{\text{br}})}$
Ensure: $c_{\text{out}} = \text{LWE}_{\vec{S}}(\frac{q}{t}f(m))$

$\text{TV} \leftarrow X^{-N/t}\sum_{i=0}^{t/2-1}\left(\frac{q}{t}f(i)X^{2Ni/t}\cdot\sum_{j=0}^{2N/t-1}X^j\right)$
$c_0, c_{\text{rem},0} \leftarrow \text{HomDivRem}(c, 2N)$
$C_0 \leftarrow \text{BlindRotate}(c_0, \text{TV}, \text{BRK}^{(B_{\text{br}}, l_{\text{br}})})$
for $i \in [K]$ **do**
  $C_i' \leftarrow \text{HomTruncRepeat}_{r_i}(C_i, [\![-T_i, T_i]\!], \beta_i)$
  $c_{i+1}, c_{\text{rem},i+1} \leftarrow \text{HomDivRem}(c_{\text{rem},i}, \beta_i)$
  $C_{i+1} \leftarrow \text{BlindRotate}(c_{i+1}, C_i', \text{BRK}^{(B_{\text{br}}, l_{\text{br}})})$
end for
$c_{\text{out}} \leftarrow \text{SampleExtract}(C_K, 0)$

3.4 Analysis of Meta-PBS

In the remainder of the paper, we assume that $2N\prod_{j\in[K]}\beta_j$ divides q. This can be ensured by modulus-switching the input LWE modulus to a proper 63-bit q. The modulus-switching noise of this step is negligible since it is orders of magnitude smaller than the input LWE noise.

3.4.1 Correctness proof. The following conditions are required to ensure the correctness of Meta-PBS.

(1) **Redundancy chain**. The initial TV must fit within R_q, and iteration i can amplify redundancy by at most β_i.

$$r_0 \le \frac{2N}{t} \text{ and } \forall i \in [K], r_{i+1} \le r_i\beta_i.$$

(2) **Sufficient final redundancy**. r_K is large enough to absorb e_K so that SampleExtract is correct.

$$r_K \ge 2\delta_K + 1.$$

(3) **HomTruncRepeat correctness**. HomTruncRepeat relies on Theorem 1 to function correctly. The variables in Theorem 1 are substituted as $[\![a,b]\!] = [\![-T_i, T_i]\!]$, $\beta = \beta_i$, $r = r_i$ and $e = e_i \in [-\delta_i, \delta_i]$.

$$\forall i \in [K], (2T_i + 1)\beta_i \le N, T_i \ge \delta_i + \lfloor \frac{r_i}{2} \rfloor.$$

(4) **Noise bound validity**. δ_i should indeed be high-probability upper bounds on $|e_i|$.

$$\Pr[|e_i| \ge \delta_i] < p_{\text{fail}}.$$

To convert Condition (4) into an explicit constraint on δ_i, we first decide the distribution of e_i using the following lemma.

Lemma 4. *If Theorem 1 holds for the first i TruncRepeats ($i \in [\![0, K]\!]$), e_i is exactly the LWE noise in* $\text{ModSwitch}(c, q/q_i')$ *and, under the Gaussian approximation above, can be modeled by*

$$\text{Var}(e_i) \le \frac{n+4}{24} + q_i'^{-2} \cdot \text{Var}(e), \quad |E(e_i)| \le \frac{1}{2}.$$

Proof. We use induction on i to prove that e_i is the noise in $\text{ModSwitch}(c, q/q_i')$.

For $i = 0$, e_0 is the LWE noise in $\text{ModSwitch}(c, 2N)$ by definition.

For $i \ge 1$, repeatedly applying Equation 4 and Equation 8 shows that $c_{\text{rem},i}$ is the component-wise centered remainder of c modulo q_i'. Applying Lemma 2 with target modulus q/q_i' and replacing $[c]^{\text{sym}}_{q_i'}$ by $c_{\text{rem},i}$ gives

$$\text{ModSwitch}(c, q/q_i') = \text{LWE}_{\vec{s},q/q_i'}\left(\frac{q}{q_i' t} m; \frac{1}{q_i'}(e - \psi^*(c_{\text{rem},i}))\right).$$

It therefore remains to prove

$$e_i = \frac{1}{q_i'}\left(e - \psi^*(c_{\text{rem},i})\right), \qquad i \in [\![0, K]\!].$$

A recurrence relation for e_i is obtained by tracking the perturbation value in each iteration. In the $(i-1)$-th iteration (Line 5-7 in Algorithm 1), HomTruncRepeat first multiplies e_{i-1} by β_{i-1} (according to Theorem 1), then the blind rotation by c_i adds $\psi^*(c_i)$ to the perturbation on the exponent. Thus,

$$\begin{aligned} e_i &= \beta_{i-1} e_{i-1} + \psi^*(c_i) \\ &= \frac{\beta_{i-1}}{q_{i-1}'}\left(e - \psi^*(c_{\text{rem},i-1})\right) + \psi^*(c_i) \\ &= \frac{1}{q_i'}\left(e - \psi^*(c_{\text{rem},i-1})\right) + \frac{1}{q_i'}\left(\psi^*(c_{\text{rem},i-1}) - \psi^*(c_{\text{rem},i})\right) \\ &= \frac{1}{q_i'}\left(e - \psi^*(c_{\text{rem},i})\right), \end{aligned}$$

where the second equality follows from the induction hypothesis, and the last two equalities use $q_{i-1}' = \beta_{i-1} q_i'$ and Equation 6. The variance of e_i follows directly from Proposition 1. □

To summarize, the four correctness conditions lead to the following theorem on the correctness of Meta-PBS.

Theorem 2. *Algorithm 1 outputs* $\text{LWE}_{\vec{S},q}(\frac{q}{t} f(m))$ *with a failure probability no greater than* $(K+1)\cdot p_{\text{fail}}$ *under the following conditions.*

(1) $r_0 \le \frac{2N}{t}$ *and* $\forall i \in [K], r_{i+1} \le r_i \beta_i$.
(2) $r_K \ge 2\delta_K + 1$.
(3) $\forall i \in [K], (2T_i + 1)\beta_i \le N, T_i \ge \delta_i + \lfloor \frac{r_i}{2} \rfloor$.
(4) $\forall i \in [K+1], \delta_i \ge \lceil \text{erfc}^{-1}(p_{\text{fail}}) \sqrt{\frac{n+4}{12} + 2q_i'^{-2} \cdot V_{\text{in}}} + \frac{1}{2} \rceil$.

Proof. We aim to prove that $M_i \cdot X^{e_i}$ is r_i-redundant in $\frac{q}{t} f(m)$. For $i = 0$, this is guaranteed by $r_0 \le \frac{2N}{t}$ and the design of TV.

For $i \ge 1$, we use induction on i. Now suppose $M_{i-1} \cdot X^{e_{i-1}}$ is r_{i-1}-redundant in $\frac{q}{t} f(m)$. Condition (4) and Condition (3) ensure that Theorem 1 holds with high probability. Combining this with Lemma 4, we know the perturbation in M_i is indeed e_i, and $M_i \cdot X^{e_i}$ is r_i-redundant in $\frac{q}{t} f(m)$ because $r_i \le r_{i-1}\beta_{i-1}$ in Condition (1). Finally, Condition (2) ensures the correctness of SampleExtract.

Correctness requires all $K+1$ perturbations $e_0, \ldots, e_K$ to satisfy their respective high-probability bounds. Applying the union bound to these $K+1$ failure events gives an overall failure probability of at most $(K+1) \cdot p_{\text{fail}}$. □

3.4.2 Noise analysis. Since HomTruncRepeat is essentially an LWE-to-RLWE packing method, it incurs an additive noise growth. That is, the output noise coefficients have the form of $e_{\text{in}} + e_{\text{add}}$, where e_{in} equals some noise coefficient in the input ciphertext, and e_{add} is independent of e_{in}.

Again, we defer the explicit noise formula until the introduction of our optimized HomTruncRepeat in Section 5. Suppose the e_{add} term introduced by the i-th HomTruncRepeat has variance $V_{\text{tr},i}$. Under the standard estimation model that the fresh noise components introduced by different blind rotations and HomTruncRepeat calls are mutually independent, the output noise variance of Meta-PBS is as follows.

Theorem 3. *The output ciphertext of Algorithm 1 has a noise variance of*

$$V_{\text{Meta-PBS}} = (K+1) \cdot V_{\text{br}} + \sum_{i \in [K]} V_{\text{tr},i}.$$

Proof. There are $K+1$ blind rotations and K packing calls. Both have additive noise growth, and the independence assumption allows their fresh-noise variances to be summed. □

Homomorphic additions and constant multiplications are carried out directly on the outputs of multiple Meta-PBS, the result of which is key-switched to the LWE secret s in preparation for the next Meta-PBS. Thus, we define c_{Hom}, the maximum capacity bits for post-bootstrap computation, as the maximum real number satisfying

$$2^{2c_{\text{Hom}}} \cdot V_{\text{Meta-PBS}} + V_{\text{ks}} \le V_{\text{in}}.$$

A larger c_{Hom} means a higher tolerance for ciphertext noise growth, thus supporting more post-bootstrap homomorphic computations.

3.4.3 Asymptotic complexity. For an RLWE dimension N and LWE dimension $n = O(N)$, evaluating a negacyclic function with $t = 2N$ in PBS/EBS requires a blind rotation with a ring dimension of $O(N^{1.5})$. The complexity of this blind rotation equals $O(\sqrt{N})$ blind rotations in R. Asymptotically, Meta-PBS can achieve the same task with a complexity of $O(1)$ blind rotations in R, given a small enough input noise.

Theorem 4. *For RLWE dimension N, LWE dimension $n = O(N)$, and $|e| < \frac{q}{4N^2}$, Meta-PBS with $K = 2$ is able to evaluate a negacyclic function with $t = 2N$ asymptotically.*

Proof. Following Lemma 3, let $aq\sqrt{N}$ be a high-probability bound on the raw phase of an LWE ciphertext $\mathbb{Z}_q^{n+1}$, where $a \geq 1$ is a constant related to p_{fail}. We set the values of r_i, δ_i and T_i as

$$r_i = \prod_{j\in[i]} \beta_j,$$
$$\delta_i = \lceil a\sqrt{N} + |e| \cdot 2N \prod_{j\in[i]} \beta_j/q \rceil,$$
$$T_i = \delta_i + \lfloor r_i/2 \rfloor.$$

These values satisfy Conditions (1), (4), and the latter half of (3) in Theorem 2, respectively. The expression of δ_i here is slightly different because the modulus-switching noise and scaled input noise in e_i are bounded separately, and q_i' is expanded.

Substituting the values of r_i, δ_i, and T_i in Condition (2) and the first half of Condition (3), we obtain the following requirements for β_0 and β_1

$$\beta_0(2\lceil a\sqrt{N} + \frac{2N}{q}|e|\rceil + 1) \leq N, \text{ for Condition (3)}$$
$$\beta_1(2\lceil a\sqrt{N} + \lfloor\beta_0/2\rfloor + \frac{2N}{q}|e|\beta_0\rceil + 1) \leq N, \text{ for Condition (3)}$$
$$\beta_0\beta_1 \geq 2\lceil a\sqrt{N} + \frac{2N}{q}|e|\beta_0\beta_1\rceil + 1, \text{ for Condition (2).}$$

The first two inequalities imply $\beta_0, \beta_1 < \sqrt{N}$. Combining this with $|e| < \frac{q}{4N^2}$, we know $\frac{4N}{q}|e|\beta_0\beta_1 < 1$. Thus, all terms related to e in the inequalities can be replaced by 1. Let $b = 2a\sqrt{N} + 5$, the inequalities can be transformed into

$$\beta_0 b \leq N,$$
$$\beta_1(b + \beta_0) \leq N,$$
$$\beta_0\beta_1 \geq b.$$

One can verify that every $(\beta_0, \beta_1) \in [\frac{N}{2b}, \frac{N}{b}] \times [\frac{2b^2}{N}, \frac{bN}{b^2+N}]$ satisfies the inequalities above. For constant a and sufficiently large N, both intervals have length $\Theta(\sqrt{N})$ and therefore contain integers, which guarantees an integer solution for β_0, β_1. □

4 Advanced LUT Evaluation

In addition to regular PBS, there exist LUT evaluation methods with extended functionalities such as PBSmanyLUT and WoP-PBS [10]. PBSmanyLUT can evaluate multiple negacyclic LUTs on the same LWE input in a single blind rotation, while WoP-PBS supports the evaluation of non-negacyclic functions. In this section, we propose Meta-PBS versions of PBSmanyLUT and WoP-PBS, which we call Meta-PBSmanyLUT and Meta-WoP-PBS, respectively.

4.1 Meta-PBSmanyLUT

4.1.1 The first attempt. The original PBSmanyLUT can evaluate 2^ν negacyclic functions modulo t in parallel. Denoting the LUT functions as f_k for $k \in [2^\nu]$, PBSmanyLUT first encodes them into a single test vector

$$\text{TV} = \sum_{k\in[2^\nu]} \text{TV}_k \cdot X^k,$$

where each TV_k encodes f_k in the same way as a regular PBS does, but in a smaller ring $R_q' = \mathbb{Z}_q[Y]/(Y^{2^{-\nu}N} + 1)$ with $Y = X^{2^\nu}$.

For $c = \text{LWE}_{\vec{s},2^{-\nu}2N}(\frac{2N\cdot 2^{-\nu}}{t}m; e)$, blind rotation in PBSmanyLUT outputs an encryption of

$$\text{TV}X^{-2^\nu\psi^*(c)} = \sum_{k\in[2^\nu]} \text{TV}_k Y^{-\psi^*(c)} X^k.$$

Here, each $\text{TV}_k \cdot Y^{-\psi^*(c)}$ is the blind-rotated test vector in the smaller ring R'. The constant term of each TV_k is extracted by $\text{SampleExtract}(\text{TV}, k)$.

However, this encoding format is incompatible with our notion of redundancy in Definition 1, preventing the adoption of our redundancy-perturbation framework.

4.1.2 An alternative TV encoding. We choose to encode f_k as

$$\text{TV} = \sum_{i\in[t/2]} \sum_{k\in[2^\nu]} X^{i\cdot 2N/t + k\cdot r_0} \frac{q}{t} f_k(i) \cdot \sum_{j\in[r_0]_{\text{sym}}} X^j,$$

where $r_0 \leq \frac{2N\cdot 2^{-\nu}}{t}$ is the initial redundancy.

Observe that the coefficient vector of $\text{TV} \cdot X^{-\min[r_0]_{\text{sym}}}$ can be divided into $t/2$ consecutive blocks of length $2N/t$, where the i-th block corresponds to

$$(\underbrace{\frac{q}{t}f_0(i), \cdots}_{r_0}, \underbrace{\frac{q}{t}f_1(i), \cdots}_{r_0}, \cdots, \underbrace{\frac{q}{t}f_{2^\nu-1}(i), \cdots}_{r_0}, 0, \cdots, 0)$$

Since the redundant coefficients are stored in a contiguous region, this arrangement offers better compatibility with our redundancy-perturbation framework, and allows us to reuse the workflow of Algorithm 1. The full algorithm is given in Algorithm 2.

We use the value $D_i = r_0 \prod_{j\in[i]} \beta_j$ for $i \in [K+1]$ to record the distance between sub-LUTs after the i-th iteration. D_K will be used to determine the extraction index for SampleExtract.

Algorithm 2 Meta-PBSmanyLUT

Require: Meta-PBS parameters from Table 1.
Require: Extra parameter ν; let $D_i = r_0 \prod_{j\in[i]} \beta_j$ for $i \in [K+1]$
Require: 2^ν negacyclic functions $\{f_k\}_{k\in[2^\nu]}$ over $\mathbb{Z}_t$
Require: Input LWE ciphertext $c = \text{LWE}_{\vec{s}}(\frac{q}{t}m; e)$ with $\text{Var}(e) \leq V_{\text{in}}$
Require: Blind rotation keys $\text{BRK}^{(B_{\text{br}}, l_{\text{br}})}$
Ensure: $c_{\text{out},k} = \text{LWE}_{\vec{S}}(\frac{q}{t}f_k(m))$ for $k \in [2^\nu]$

$\text{TV} \leftarrow \sum_{i\in[t/2]} \sum_{k\in[2^\nu]} X^{i\cdot 2N/t + k\cdot r_0} \frac{q}{t} f_k(i) \cdot \sum_{j\in[r_0]_{\text{sym}}} X^j$
$c_0, c_{\text{rem},0} \leftarrow \text{HomDivRem}(c, 2N)$
$C_0 \leftarrow \text{BlindRotate}(c_0, \text{TV}, \text{BRK}^{(B_{\text{br}}, l_{\text{br}})})$
for $i \in [K]$ **do**
  $C_i' \leftarrow \text{HomTruncRepeat}_{r_i}(C_i, [\![-T_i, T_i + D_i(2^\nu - 1)]\!], \beta_i)$
  $c_{i+1}, c_{\text{rem},i+1} \leftarrow \text{HomDivRem}(c_{\text{rem},i}, \beta_i)$
  $C_{i+1} \leftarrow \text{BlindRotate}(c_{i+1}, C_i', \text{BRK}^{(B_{\text{br}}, l_{\text{br}})})$
end for
for $k \in [2^\nu]$ **do**
  $c_{\text{out},k} \leftarrow \text{SampleExtract}(C_K, k \cdot D_K)$
end for

Theorem 5. *Algorithm 2 outputs* $\{\text{LWE}_{\vec{S}}(\frac{q}{t}f_k(m))\}_{k\in[2^\nu]}$ *with a failure probability no more than* $(K+1)\cdot p_{\text{fail}}$ *under the following conditions.*

(1) $r_0 \leq \frac{2N\cdot 2^{-\nu}}{t}$ *and* $\forall i \in [K], r_{i+1} \leq r_i \cdot \beta_i$.

(2) $r_K \geq 2\delta_K + 1$.
(3) $\forall i \in [K], (2T_i + 1 + (2^\nu - 1)D_i) \cdot \beta_i \leq N, T_i \geq \delta_i + \lfloor \frac{r_i}{2} \rfloor$.
(4) $\forall i \in [K+1], \delta_i \geq \lceil \mathrm{erfc}^{-1}(p_{\mathrm{fail}}) \cdot \sqrt{\frac{n+4}{12} + 2q_i'^{-2} \cdot V_{\mathrm{in}}} + \frac{1}{2} \rceil$.

Theorem 5 presents the correctness conditions of Algorithm 2. The proof is similar to that of Meta-PBS. The main difference is that we aim to prove $M_i \cdot X^{e_i} \cdot X^{-kD_i}$ is r_i-redundant in $\frac{q}{t} f_k(m)$ for $k \in [2^\nu]$. The base case is that $\mathrm{TV} \cdot X^{-u \cdot 2N/t - kD_0}$ is D_0-redundant in $\frac{q}{t} f_k(u)$ for $u \in [t/2]$ and $k \in [2^\nu]$. The initial bound on r_0 places all 2^ν sub-LUTs in one message block, while iteration i multiplies each sub-LUT's redundancy and the distance between sub-LUTs by β_i. Notably, the perturbation bounds are the same as in Meta-PBS because the initial HomDivRem still targets modulus $2N$; in particular, they do not contain the $2^{-\nu}$ factor used by the original PBSmanyLUT rotation.

The output-noise formula remains that of Theorem 3, but $V_{\mathrm{tr},i}$ must be evaluated for the enlarged interval $[\![-T_i, T_i + D_i(2^\nu - 1)]\!]$. For the optimized construction in Lemma 6, this means substituting $a = -T_i$ and $b = T_i + D_i(2^\nu - 1)$.

4.2 Meta-WoP-PBS

To enable evaluation of an arbitrary function $f : \mathbb{Z}_t \to \mathbb{Z}_t$ within the Meta-PBS framework, we adapt the FDFB-CancelSign approach [20] (which can be viewed as a variant of WoP-PBS) by modifying its key steps to fit Meta-PBS.

We begin by briefly reviewing the steps of FDFB-CancelSign. Given an LWE ciphertext $c = \mathrm{LWE}(\frac{q}{t} m) \in \mathbb{Z}_q^{n+1}$, FDFB-CancelSign homomorphically computes $f(m)$ using two blind rotations and one LWE-to-RLWE packing as follows.

(1) It first extends f to a negacyclic function f^* modulo $2t$. The extended function $f^* : \mathbb{Z}_{2t} \to \mathbb{Z}_t$ satisfies $f^*(i) = f(i)$ for $i \in [t]$. Meanwhile, c is lifted to $2q$, producing $c' = \mathrm{LWE}_{\vec{s},2q}(\frac{q}{t} m + q\gamma)$, where $\gamma \in \{0, 1\}$ is an unknown bit introduced by this step.
(2) Then, it performs PBSmanyLUT using c' to obtain $c_{\mathrm{sgn}} = \mathrm{LWE}(\frac{q}{2}\gamma)$ and a LWE encryption of $\frac{q}{t} f^*(m + \gamma t) = \frac{q}{t} f(m) \cdot (-1)^\gamma$.
(3) It packs $\frac{q}{t} f(m) \cdot (-1)^\gamma$ redundantly into an RLWE ciphertext and blind-rotates the RLWE ciphertext by c_{sgn}. The resulting ciphertext stores $\frac{q}{t} f(m)$ at the constant term since the blind rotation cancels the $(-1)^\gamma$ term.
(4) Extracting the constant term finishes the bootstrapping.

Meta-WoP-PBS is realized by replacing PBSmanyLUT with Meta-PBSmanyLUT in step (2) and instantiating the LWE-to-RLWE packing with HomTruncRepeat in step (3). Setting $\nu > 1$ in Meta-PBSmanyLUT evaluates multiple LUTs in one Meta-WoP-PBS; step (4) then cancels the $(-1)^\gamma$ terms of the resulting LWE ciphertexts in parallel.

The full algorithm and its correctness conditions are given in Appendix C. The output noise variance of Meta-WoP-PBS, denoted $V_{\mathrm{Meta-WoP-PBS}}$, is dominated by $V_{\mathrm{Meta-PBS}} + V_{\mathrm{br}}$ from the Meta-PBSmanyLUT and final blind-rotation stages. Its exact formula, including the HomTruncRepeat noise, is given in Corollary 2.

4.3 Homomorphic Linear Combinations after Meta-WoP-PBS

As pointed out by Bergerat et al., a typical component in a TFHE circuit consists of a homomorphic linear combination, a key-switching, and a PBS [3]. This motivates us to investigate the strength of Meta-WoP-PBS in supporting homomorphic linear combinations.

To support large-scale homomorphic linear combinations, ciphertexts are typically instantiated as Lev ciphertexts to mitigate noise growth from integer–ciphertext multiplications. Meta-WoP-PBS can be configured to output Lev ciphertexts with a gadget length $l_{\mathrm{msg}} = 2^\nu - 1$ to address this requirement. In this setting, the noise variance after a length L-linear combination is

$$L \cdot \max_{v \in \mathbb{Z}_t} \left(\left\| G^{(B_{\mathrm{msg}}, l_{\mathrm{msg}}, t)}(v) \right\|_2^2 \right) \cdot V_{\mathrm{Meta-WoP-PBS}}.$$

Taking $t = 2^8$ as an example, the noise-variance growth factor for $\mathbb{Z}_t$-ciphertext multiplication is 2^{14} when $\nu = 1$, and is reduced to $4^2 + 4^2 + 2^2 = 36$ for $\nu = 2$.

Remark. Using a Lev ciphertext in place of a single LWE ciphertext also entails drawbacks. In particular, because Meta-WoP-PBS needs l_{msg} gadget-component LUTs and one sign LUT, the number of distinct LUT entries grows from $2t$ to $(l_{\mathrm{msg}} + 1)t$. The maximal value of t therefore drops from $\frac{N}{2}$ to $\frac{N}{l_{\mathrm{msg}}+1}$, limiting its applicability at higher precision.

5 Optimizing HomTruncRepeat

5.1 Existing approaches

Since HomTruncRepeat extracts LWE ciphertexts from an RLWE ciphertext and repacks them, it can be evaluated using PackRLWE, as discussed in Section 2.4.

Applying the base-aware packing optimization allows us to absorb the factor $\sum_{j \in [\beta]} X^j \cdot X^{\Delta_{r,\beta}}$ of Definition 2 into the public keys when computing $\mathrm{HomTruncRepeat}_r(M, [\![a, b]\!], \beta)$. Therefore, we focus our discussion on the remaining part, which we denote as $\mathrm{TruncPad}(M, [\![a, b]\!], \beta)$. Its output has the coefficient vector

$$\underbrace{(M[0], 0, \cdots, 0}_{\beta \text{ coefficients}}, \cdots, \underbrace{M[b], 0, \cdots, 0}_{\beta \text{ coefficients}}, 0, 0, \cdots$$
$$\cdots, 0, \underbrace{M[N+a], 0, \cdots, 0}_{\beta \text{ coefficients}}, \cdots, \underbrace{M[N-1], 0, \cdots, 0)}_{\beta \text{ coefficients}}.$$

TruncPad can be homomorphically evaluated as

$$\begin{aligned} &\mathrm{HomTruncPad}(C = (A, B), [\![a, b]\!], \beta) \\ &= \sum_{i \in [N]} \mathrm{TruncPad}(A \cdot X^i, [\![a, b]\!], \beta) \odot \mathrm{RLev}(S[i]) \\ &\quad + \mathrm{TruncPad}(B, [\![a, b]\!], \beta). \end{aligned}$$

Equation 9 shows the computation of $\mathrm{HomTruncPad}((A, B) = \mathrm{RLWE}_S(M; E), [\![-1, 1]\!], 2)$ with $N = 8$ and $M' = M + E$. Ciphertexts are marked in boxes. Column vectors are coefficient vectors of plaintext polynomials.

$$\begin{pmatrix} A[0]\\0\\A[1]\\0\\0\\0\\A[7]\\0 \end{pmatrix}\cdot\boxed{S[0]}+\begin{pmatrix} -A[7]\\0\\A[0]\\0\\0\\0\\A[6]\\0 \end{pmatrix}\cdot\boxed{S[1]}+\cdots\begin{pmatrix} -A[1]\\0\\-A[2]\\0\\0\\0\\A[0]\\0 \end{pmatrix}\cdot\boxed{S[7]}+\begin{pmatrix} B[0]\\0\\B[1]\\0\\0\\0\\B[7]\\0 \end{pmatrix}=\boxed{\begin{pmatrix} M'[0]\\0\\M'[1]\\0\\0\\0\\M'[7]\\0 \end{pmatrix}} \quad (9)$$

HomTruncRepeat is then realized by replacing $\mathrm{RLev}(S[i])$ in the TruncPad evaluation with $\mathrm{RLev}(S[i]\cdot\sum_{j\in[\beta]}X^j\cdot X^{\Delta_{r,\beta}})$. This realization of HomTruncRepeat has a computational cost of $N^2\log N\, l_{\mathrm{tr}}$ $\mathbb{Z}_q$-multiplications and a storage cost of $2N^2 l_{\mathrm{tr}}$ $\mathbb{Z}_q$-integers in the key-switching key, which is comparable to the cost of a blind rotation. In the following, we show how to significantly reduce the noise variance, computational cost, and storage cost of TruncRepeat.

5.2 Compact HomTruncRepeat from locally negacyclic matrix.

From the coefficient vector point of view, the multiplication of two polynomials U, V is equivalent to left-multiplying $\vec{V}$ with the negacyclic matrix of U. Therefore, by merging multiple plaintext coefficient vectors from the LHS of Equation 9 into one negacyclic matrix, multiple polynomial-RLev products can be replaced by a single one. The modified procedure is denoted by HomTruncPad*.

The following is an example of HomTruncPad*, where every two columns in Equation 9 are merged into a negacyclic matrix. Matrix entries marked by $*$ and $\cdots$ are constructed via negacyclicity. A side effect of this negacyclic extension is the appearance of "garbage" values in the unused rows (marked in bold red).

$$\begin{pmatrix} A[0] & * & -A[7] & \cdots\\ 0 & * & 0 & \cdots\\ A[1] & * & A[0] & \cdots\\ 0 & * & 0 & \cdots\\ \mathbf{A[6]} & * & \mathbf{A[1]} & \cdots\\ 0 & * & 0 & \cdots\\ A[7] & * & A[6] & \cdots\\ 0 & * & 0 & \cdots \end{pmatrix}\cdot\boxed{\begin{pmatrix} S[0]\\0\\S[1]\\0\\0\\0\\0\\0 \end{pmatrix}}+\begin{pmatrix} -A[6] & * & -A[5] & \cdots\\ 0 & * & 0 & \cdots\\ -A[7] & * & -A[6] & \cdots\\ 0 & * & 0 & \cdots\\ \mathbf{A[4]} & * & \mathbf{-A[7]} & \cdots\\ 0 & * & 0 & \cdots\\ A[5] & * & A[4] & \cdots\\ 0 & * & 0 & \cdots \end{pmatrix}\cdot\boxed{\begin{pmatrix} S[2]\\0\\S[3]\\0\\0\\0\\0\\0 \end{pmatrix}}+\cdots$$

$$\cdots+\begin{pmatrix} -A[2] & * & -A[1] & \cdots\\ 0 & * & 0 & \cdots\\ -A[3] & * & -A[2] & \cdots\\ 0 & * & 0 & \cdots\\ \mathbf{A[0]} & * & \mathbf{-A[3]} & \cdots\\ 0 & * & 0 & \cdots\\ A[1] & * & A[0] & \cdots\\ 0 & * & 0 & \cdots \end{pmatrix}\cdot\boxed{\begin{pmatrix} S[6]\\0\\S[7]\\0\\0\\0\\0\\0 \end{pmatrix}}+\begin{pmatrix} B[0]\\0\\B[1]\\0\\0\\0\\B[7]\\0 \end{pmatrix}=\begin{pmatrix} M'[0]\\0\\M'[1]\\0\\\mathbf{Garbage}\\0\\M'[7]\\0 \end{pmatrix}$$

More generally, HomTruncPad$^*(M, [\![a,b]\!], \beta, \epsilon)$ merges $\epsilon+1$ consecutive columns and outputs a polynomial with a coefficient vector of

$$(\underbrace{M[0],0,\cdots,0}_{\beta},\cdots,\underbrace{M[b],0,\cdots,0}_{\beta},\underbrace{*,\cdots,*}_{\epsilon\beta\text{ garbage}},0,\cdots$$
$$\cdots,0,\underbrace{*,\cdots,*}_{\epsilon\beta\text{ garbage}},\underbrace{M[N+a],0,\cdots,0}_{\beta},\cdots,\underbrace{M[N-1],0,\cdots,0}_{\beta})$$

Thus we require $(b-a+1+\epsilon)\beta \le N$ to prevent the garbage coefficients from overlapping with the useful ones.

A formal definition of HomTruncPad* and HomTruncRepeat* is given in Definition 4. Replacing TruncRepeat by TruncRepeat* in Meta-PBS modifies the correctness conditions as stated in Lemma 5. Its proof is given in Appendix D.

Definition 4. *For $\epsilon \in \mathbb{Z}_{\ge 0}$,* HomTruncPad$^*((A,B), [\![a,b]\!], \beta, \epsilon)$ *is defined as*

$$\mathrm{TruncPad}(B, [\![a,b]\!], \beta) + \sum_{j\in[\lceil N/(\epsilon+1)\rceil]} \mathrm{TruncPad}(A\cdot X^{j(\epsilon+1)}, [\![a-\epsilon, b]\!], \beta) \odot \mathrm{RLev}\Big(\sum_{i\in[\![j(\epsilon+1),\min(j(\epsilon+1)+\epsilon, N-1)]\!]} S[i]\cdot X^{\beta(i-j(\epsilon+1))}\Big).$$

Again, HomTruncRepeat$^*_r((A,B), [\![a,b]\!], \beta, \epsilon)$ *is obtained in the base-aware packing way, by multiplying the* RLev *ciphertexts used in* HomTruncPad* *by* $\sum_{k\in[\beta]} X^k\cdot X^{\Delta_{r,\beta}}$.

Lemma 5. *By replacing the i-th* HomTruncRepeat$(\cdots)$ *in Algorithm 1 by* HomTruncRepeat$^*(\cdots, \epsilon_i)$*, the algorithm outputs* $\mathrm{LWE}_{\vec{S},q}(\frac{q}{t}f(m))$ *with failure probability at most* $(K+1)\cdot p_{\mathrm{fail}}$ *under Conditions 1, 2, and 4, together with the modified Condition 3:*

$$\forall i\in[K], (2T_i+1+\epsilon_i)\beta_i \le N, T_i \ge \delta_i + \lfloor \frac{r_i}{2} \rfloor.$$

The noise growth of HomTruncRepeat/HomTruncRepeat* is characterized in Lemma 6. Its proof is given in Appendix D. $V_{\mathrm{tr},i}$ in $V_{\mathrm{Meta-PBS}}$ equals $V_{\mathrm{fill}}(\epsilon_i)$.

Lemma 6. *Let* $C = \mathrm{RLWE}_S(M;E)$ *with* $\mathrm{Var}(E[i]) = \sigma^2$ *for* $i \in [\![a,b]\!]_N$*, and let integers* $a \le 0 \le b$, $\beta > 0$, $\epsilon \ge 0$ *satisfy* $(b-a+1+\epsilon)\beta \le N$*. Denote* $C' = \mathrm{HomTruncRepeat}^*_r(C, [\![a,b]\!], \beta, \epsilon) = \mathrm{RLWE}_S(\mathrm{TruncRepeat}^*_r(M, [\![a,b]\!], \beta, \epsilon); E')$*. Let* $B_{\mathrm{tr}}, l_{\mathrm{tr}}$ *be the gadget parameters used in* HomTruncRepeat* *and let* σ_{trk} *be the standard deviation of the noise in the RLev ciphertexts. Then, for each index* $i \in [N]$ *corresponding to the garbage values,*

$$V_{\mathrm{emp}}(\epsilon) := \mathrm{Var}(E'[i]) = (N+(b-a)\lceil\frac{N}{\epsilon+1}\rceil) l_{\mathrm{tr}}(B_{\mathrm{tr}}^2+2)(\frac{\sigma_{\mathrm{trk}}^2}{12}+2^{19.4}\cdot N).$$

For each index $i \in [N]$ *corresponding to the useful values,*

$$V_{\mathrm{fill}}(\epsilon) := \mathrm{Var}(E'[i]) - \sigma^2 = \frac{N}{2}\Big(\frac{q^2}{12B_{\mathrm{tr}}^{2l_{\mathrm{tr}}}} - \frac{1}{12}\Big) + \frac{N}{16} + V_{\mathrm{emp}}(\epsilon).$$

5.3 Comparing with other packing methods.

We compare HomTruncRepeat* with existing packing methods in terms of computational cost, evaluation key size, noise growth, and functionality when evaluating $\mathrm{TruncRepeat}(\cdot, [\![-T_i, T_i]\!], \beta_i)$.

PubFuncKS [8] and base-aware PackingKS [13]. According to Section 5.1, the HomTruncRepeat instantiated via base-aware PackingKS is a special case of HomTruncRepeat* with $\epsilon = 0$. Hence, compared with base-aware PackingKS, HomTruncRepeat* decreases the variance term $V_{\mathrm{emp}}(\epsilon)$ by approximately a factor of $\epsilon+1$. In addition, due to the column-merging technique, both the computational cost and the evaluation key size are reduced by roughly the same factor.

Automorphism-based packing [6, 17]. The automorphism-based packing methods are favorable for their small evaluation key size (i.e., only $\log N$ keys) and small additive noise growth of $\log N$ key-switchings. However, they are designed for power-of-two β and only compute HomTruncPad$(\cdot, [\![-T_i, T_i]\!], \beta_i)$ instead of HomTruncRepeat, leading to additional problems.

For a power-of-two β_i, these methods use $2T_i + 1 + \log(N/\beta_i)$ automorphisms and add the noise of $\log N$ key switchings. The

resulting ciphertext must be multiplied by $\sum_{j\in[\beta_i]} X^j$ to obtain the HomTruncRepeat result, amplifying the noise variance by β_i. Restricting β_i to powers of two also greatly limits the parameter search space, potentially increasing the number of required iterations.

For non-power-of-two β_i, all the $\beta_i(2T_i+1)$ coefficients of the TruncRepeat result have to be packed using a packing base of one. This costs no less than $\beta_i(2T_i+1)$ automorphisms and introduces the noise variance of $\log N$ key-switchings.

For the parameters used in our experiments, automorphism-based packing methods have a much smaller evaluation key size, a higher noise variance, and a higher computational cost than HomTruncRepeat*. Appendix E gives the quantitative cost, key-size, and noise comparison.

6 Implementation

6.1 Parameter selection

Meta-PBS has many parameters that have a complicated impact on the overall performance. For example, choosing the largest β_i greedily in order may seem to be a good option to obtain the lowest K, thus decreasing the number of required blind rotations. However, large β_i's in the early iterations will lead to the fast growth of r_i, δ_i, and T_i, which will force us to use small β_i's in later iterations, resulting in non-optimal K.

Given the difficulty of selecting optimal parameters, we propose a heuristic parameter selection algorithm to automate this process. It enumerates packing bases in reverse order from the target final redundancy, filters candidates using the correctness and noise constraints, and selects the valid candidate with the smallest estimated FFT cost. The pseudocode and full details are given in Appendix F.

6.2 Comparing with other LUT evaluation approaches

We compare Meta-PBS with EBS and integer-input LHE in their efficiency when evaluating large LUTs and their capability to perform $\mathbb{Z}$-linear combinations after bootstrapping. All experiments are conducted single-threaded on a workstation with Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz and Fedora release 38. Our implementation is developed on TFHE-go[3] [14]. All LWE/RLWE parameters (including those in existing methods) are decided or updated to meet 130 bits of security using the Lattice Estimator[4] [1] with `red_cost_model=BDGL16`, in alignment with TFHE-rs [27]. The timing results are averaged over 1000 runs unless otherwise specified. Following [25], the compressed evaluation key size is reported, i.e., all the encryption randomness in LWE/RLWE ciphertexts is replaced by a single random seed. We experimentally validate the noise analysis in Appendix H. All reported Meta-PBS and Meta-WoP-PBS benchmarks, parameter searches, and failure-probability estimates use the randomized blind-rotation mitigation described there. The mitigation centers the implementation-dependent biased output noise so that the centered-noise failure estimates apply. Its measured latency overhead is negligible because it adds only lightweight negacyclic rotations of the accumulator.

[3] https://github.com/sp301415/tfhe-go/tree/2369903
[4] https://github.com/malb/lattice-estimator/tree/5ba00f5

6.2.1 Comparing with EBS [16]. The experiments are limited to negacyclic PBS because the current EBS implementation does not support WoP-PBS. For each $\log_2 t \in [\![8, 12]\!]$, we first select the fastest EBS parameter set subject to $c_{\mathrm{Hom}} > 0.5$, i.e., enough capacity for at least one homomorphic addition. We then configure Meta-PBS to provide a c_{Hom} no smaller than that of EBS with the same t. Following the default settings in TFHE-go, we set the overall failure probability of EBS to 2^{-64}. For Meta-PBS, we set the per-perturbation failure probability to $p_{\mathrm{fail}} = 2^{-66}$, then the union bound in Theorem 2 gives an overall failure probability of at most 2^{-64} for all parameter sets below. The parameters are given in Table 2 and Table 3.

Table 2: Security parameters for EBS and Meta-PBS.

n	N	q	σ_{LWE}/q	σ_{RLWE}/q	Overall fail prob
1170	2048	2^{64}	$2^{-27.44}$	$2^{-50.22}$	2^{-64}

The benchmark results are given in Table 4. As expected, the running time of EBS increases linearly with t, while the running time of Meta-PBS grows more smoothly, leading to a speedup of 3.1–81.0×. Meta-PBS is slower when $t = 2^{11}$ than when $t = 2^{12}$ because the former uses a more expensive parameter set to match the higher c_{Hom} target of EBS. However, Meta-PBS incurs a larger evaluation key size due to the additional HomTruncRepeat keys, and this overhead becomes more significant when the ϵ_i's are small. For the parameter sets in Table 4, the compressed evaluation key grows from about 73 MB for EBS to 74–112 MB for Meta-PBS. In the IND-CPA^D security model, bootstrapping failures can be exploited in key-recovery attacks. We therefore also evaluate Meta-PBS with each perturbation-bound event set to 2^{-130}, so that the overall failure probability is at most 2^{-128}. The parameter sets use the same LWE/RLWE security parameters as Table 2, with their main parameters and performance given in Table 5 and Table 6.

Compared with the parameter sets whose overall failure probability is at most 2^{-64}, these stricter Meta-PBS parameter sets take 154–271 ms and require 73.44–111.38 MB of evaluation keys. The

Table 3: Main parameters for EBS and Meta-PBS.

t	Method	k^*	B_{br}	l_{br}	B_{ks}	l_{ks}	c_{Hom}	K	B_{tr}	l_{tr}
2^8	EBS	2^3	2^{15}	2	2^5	5	3.92	/	/	/
	Meta	1	2^{15}	2	2^9	2	5.25	1	2^{23}	1
2^9	EBS	2^4	2^{15}	2	2^4	6	2.81	/	/	/
	Meta	1	2^{15}	2	2^7	3	4.24	2	2^{23}	1
2^{10}	EBS	2^5	2^{15}	2	2^3	8	1.46	/	/	/
	Meta	1	2^{15}	2	2^7	3	1.73	2	2^{23}	1
2^{11}	EBS	2^7	2^{15}	2	2^7	3	6.11	/	/	/
	Meta	1	2^{11}	3	2^4	6	6.38	2	2^{15}	2
2^{12}	EBS	2^8	2^{15}	2	2^7	3	3.77	/	/	/
	Meta	1	2^{15}	2	2^5	5	3.81	2	2^{15}	2

* Blind rotation is performed in a ring dimension of kN.

Table 4: Performance of EBS and Meta-PBS.

t	Method	Time (ms)	Speedup	c_{Hom}	Evaluation key size (MB)
2^8	EBS	331	1x	3.92	73.16
	Meta	107	3.1x	5.25	73.54
2^9	EBS	670	1x	2.81	73.17
	Meta	157	4.3x	4.24	73.36
2^{10}	EBS	1,323	1x	1.46	73.18
	Meta	160	8.3x	1.73	73.47
2^{11}	EBS*	5,368	1x	6.11	73.15
	Meta	209	25.7x	6.38	111.04
2^{12}	EBS*	13,442	1x	3.77	73.15
	Meta	166	81.0x	3.81	80.44

* Timing is averaged over 100 instead of 1000 runs.

Table 5: Main Meta-PBS parameters with overall failure probability at most 2^{-128}.

t	B_{br}	l_{br}	B_{ks}	l_{ks}	c_{Hom}	K	B_{tr}	l_{tr}
2^8	2^{15}	2	2^9	2	5.60	2	2^{23}	1
2^9	2^{15}	2	2^7	3	3.30	2	2^{23}	1
2^{10}	2^{15}	2	2^5	5	3.02	2	2^{15}	2
2^{11}	2^{11}	3	2^5	5	6.37	3	2^{15}	2
2^{12}	2^{11}	3	2^5	5	6.31	3	2^{15}	2

Table 6: Meta-PBS performance with overall failure probability at most 2^{-128}.

t	Time (ms)	c_{Hom}	Evaluation key size (MB)
2^8	154	5.60	73.44
2^9	156	3.30	73.49
2^{10}	158	3.02	74.63
2^{11}	271	6.37	110.60
2^{12}	270	6.31	111.38

stricter bound primarily increases the required perturbation margin; for 11- and 12-bit plaintexts it also increases the iteration count from two to three.

6.2.2 Comparing with integer-input LHE_8 [25]. We compare Meta-WoP-PBS at $t = 2^8$ with the LHE_8 parameter set of integer-input LHE[5], which takes one 8-bit ciphertext as input. Meta-WoP-PBS is configured with $\nu = 2$ and supports a maximum post-bootstrap linear combination size $L = \lfloor 2^{2c_{\text{Hom}}}/36 \rfloor$. For LHE_8, a post-bootstrap scalar multiplication is evaluated by constructing a test vector that encodes this function, before computing external products between that test vector and the eight RGSW ciphertexts encrypting the message bits. An L-sized linear combination is realized by adding the outputs of L scalar multiplications. We compute L according to Appendix E of [25] and define $c_{\text{Hom}} = \log_2(L)/2$.

[5]https://github.com/KAIST-CryptLab/refined-tfhe-lhe/tree/9b0426e

The regular parameter sets are given in Table 7 and Table 8. Meta_8 and Meta_8C provide different tradeoffs between computation time and L: Meta_8 supports a larger L, whereas Meta_8C sacrifices capacity for lower latency. The Int-LHE parameter sets LHE_4_4 and LHE_8, corresponding respectively to two 4-bit input ciphertexts and one 8-bit input ciphertext, are taken from Table 7 of [25]; LHE_4_4 is compared later with other decomposition-based methods. Both Int-LHE sets have overall failure probability 2^{-40}. For Meta_8 and Meta_8C, the per-perturbation bound is 2^{-42}; with $K = 2$, the union bound over $K + 2 = 4$ events is also at most 2^{-40}.

We use the published TFHE-rs implementation for the Int-LHE-specific Conv. stage, but remeasure the four Extr.+Refr. operations in TFHE-go to reduce timing differences between libraries. For LHE_8, these four operations take 4×74.4 ms in TFHE-go, compared with 4×64.7 ms in TFHE-rs; the table uses the TFHE-go measurement.

All parameter sets use $N = 2048$ and $q = 2^{64}$. As shown in the table, Meta_8 and Meta_8C achieve a 2.04–2.61× speedup over LHE_8. This improvement comes from two main factors. First, the additive noise growth of our method enables the choice of $l_{\text{br}} = 2$ or 3, whereas LHE_8 must use $l_{\text{br}} = 4$ to control the $O(N)$ multiplicative noise growth factor introduced by the Conv. operation. Since both methods require four blind rotations, the smaller l_{br} in our method reduces the overall cost of blind rotation. Second, in our method, the running time is dominated by blind rotations, while all other operations take only about 30 ms in total. By contrast,

Table 7: Security parameters for Int-LHE and Meta-WoP-PBS.

Method	n	σ_{LWE}/q	σ_{RLWE}/q	Failure
LHE_4_4	769	$2^{-17.09}$	$2^{-50.22}$	overall 2^{-40}
LHE_8	953	$2^{-21.84}$	$2^{-50.22}$	overall 2^{-40}
Meta_8C Meta_8	970	$2^{-22.28}$	$2^{-50.22}$	per event 2^{-42}; overall $\leq 2^{-40}$
Meta_8C-128 Meta_8-128	970	$2^{-22.28}$	$2^{-50.22}$	per event 2^{-130}; overall $\leq 2^{-127.68}$

Table 8: Main parameters for Int-LHE and Meta-WoP-PBS.

Method	B_{br}	l_{br}	B_{ks}	l_{ks}	c_{Hom}
LHE_4_4	2^{15}	2	2^4	3	NA
LHE_8	2^9	4	2^7	2	4.91
Meta_8C	2^{15}	2	2^2	10	6.68
Meta_8	2^{11}	3	2^2	10	9.89
Meta_8C-128	2^{11}	3	2^2	10	8.03
Meta_8-128	2^9	4	2^2	10	10.42

Method	K	B_{tr}	l_{tr}	ν
Meta_8C	2	2^{11}	3	2
Meta_8	2	2^{11}	3	2
Meta_8C-128 Meta_8-128	3	$2^{12}, 2^{11}, 2^{11}$	3	2

Table 9: Performance of LHE_8 and Meta-WoP-PBS.

Method	Time* (ms)	Speedup	L	Key size (MB)
LHE_8	543	1x	905	120.45
Meta_8C	208	2.61x	> 290	70.74
Meta_8	266	2.04x	> 25,000	100.21
Meta_8C-128	307	–	> 1,800	99.97
Meta_8-128	372	–	> 50,000	130.29

* The time for performing linear combinations is not included. No speedup is reported for the two "-128" sets because there is no Int-LHE baseline with the same failure probability.

LHE_8 additionally requires Conv. operations, whose running time is comparable to the four blind rotations.

Additive noise growth also lets Meta_8 support a much larger L than LHE_8. Both Meta_8 and Meta_8C also use smaller evaluation keys than LHE_8.

For both "-128" sets, each perturbation-bound event has probability 2^{-130}. Because $K = 3$ gives $K + 2 = 5$ events, their union bound is $5 \cdot 2^{-130} \approx 2^{-127.68}$, which is close to but slightly above 2^{-128}. Compared with their 2^{-40} counterparts, Meta_8C-128 and Meta_8-128 take 307 ms and 372 ms and use 99.97 MB and 130.29 MB of evaluation keys, respectively.

6.2.3 Comparison with LargeLUT.. Following the scope discussion in the introduction, we compare only the core single-ciphertext LUT-evaluation phase of LargeLUT and Meta-PBS.

Because the LargeLUT implementation is not public, we estimate the latency of its fastest reported 12-bit configuration by counting its operations and measuring the corresponding primitives on the same platform and with the same security parameters as Meta-PBS. The gadget parameters are obtained using the Meta-PBS parameter-selection program. Four blind rotations take approximately 212 ms (4×53.0 ms), 624 SampleExtract-with-KeySwitch operations take approximately 3,114 ms (a measured batch of 12 at 59.9 ms, scaled to 52 batches), and three PackingKS operations take approximately 180 ms (3×60.1 ms), giving a total estimate of 3,507 ms. The corresponding 12-bit Meta-PBS benchmark takes 166 ms; the estimated core-phase latency of LargeLUT is therefore about 21.1× that of Meta-PBS. This comparison is an operation-level estimate of the core phase. It excludes LargeLUT's input preprocessing and is not an end-to-end benchmark. The operation-level estimate is summarized in Table 10.

6.2.4 Comparison with decomposed-input methods. The following results provide timings for 8-bit arbitrary LUT evaluation in both Meta-PBS and decomposed-input methods. We restress that this is not an end-to-end comparison because the methods use different input and output representations.

LHE_4_4 represents an 8-bit input using two 4-bit ciphertexts and evaluates an arbitrary 8-to-8-bit LUT in 198 ms, compared with 208 ms for Meta_8C. Its evaluation key is 49.33 MB, whereas that of Meta_8C is 70.74 MB. Since LHE_4_4 produces a digit-decomposed output, the single-LWE linear-combination measures c_{Hom} and L do not apply to this configuration.

For the tree-based method of Trama et al. [24], the reported 8-bit configuration uses two 4-bit input ciphertexts, decomposition base $B_{\mathrm{tree}} = 2^4$, three blind rotations, and two public functional key switchings. Based on same-platform primitive measurements, we estimate its latency as $3 \times 58.3 + 2 \times 123.5 \approx 422$ ms and its evaluation key size as approximately 160 MB.

The results are summarized in Table 11.

At these respective parameter sets, the estimated tree-based latency is 2.13× that of LHE_4_4. It is 1.59× and 2.03× that of Meta_8 and Meta_8C, respectively. These latency ratios do not compare equal failure probabilities. The tree-based estimate uses 2^{-23}, whereas LHE_4_4 and two Meta configurations use an overall failure probability of 2^{-40}. For bitwise operations, the tree-based method requires only two blind rotations and has an estimated latency below 120 ms, illustrating that its relative performance depends on the workload.

6.2.5 Discussion. Taken together, the results show that the measured advantage of Meta-PBS is concentrated in evaluating a LUT from one ciphertext and then performing substantial linear computation. Methods using decomposed inputs retain advantages when the input is already decomposed or the workload is primarily bitwise. The latency ratios above should therefore be interpreted under their stated input representations, output formats, and failure-probability settings.

Acknowledgments

This work was supported by the National Cyber Security-National Science and Technology Major Project (2026ZD1500900), the Scientific Research Innovation Capability Support Project for Young Faculty (ZYGXQNJSKYCXNLZCXM-P4), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM114), and Ant Group Research Fund (AJL2503730006).

We thank ANT group.

References

[1] Martin R. Albrecht, Rachel Player, and Sam Scott. 2015. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology* 9, 3 (2015), 169–203. doi:10.1515/jmc-2015-0016

[2] Youngjin Bae, Jung Hee Cheon, Wonhee Cho, Jaehyung Kim, and Taekyung Kim. 2022. META-BTS: Bootstrapping Precision Beyond the Limit. In *ACM CCS 2022: 29th Conference on Computer and Communications Security*, Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi (Eds.). ACM Press, Los Angeles, CA, USA, 223–234. doi:10.1145/3548606.3560696

[3] Loris Bergerat, Anas Boudi, Quentin Bourgerie, Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. 2023. Parameter Optimization and Larger Precision for (T)FHE. *Journal of Cryptology* 36, 3 (July 2023), 28. doi:10.1007/s00145-023-09463-5

[4] Loris Bergerat, Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, Adeline Roux-Langlois, and Samuel Tap. 2024. New Secret Keys for Enhanced Performance in (T)FHE. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New York, NY, USA, 2547–2561. doi:10.1145/3658644.3670376

[5] Christina Boura, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev. 2019. Simulating Homomorphic Evaluation of Deep Learning Predictions. In *Cyber Security Cryptography and Machine Learning*, Shlomi Dolev, Danny Hendler, Sachin Lodha, and Moti Yung (Eds.). Springer International Publishing, Cham, 212–230.

[6] Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. 2021. Efficient Homomorphic Conversion Between (Ring) LWE Ciphertexts. In *ACNS 2021: 19th International Conference on Applied Cryptography and Network Security, Part I (Lecture Notes in Computer Science, Vol. 12726)*, Kazue Sako and Nils Ole Tippenhauer (Eds.).

Table 10: Operation-level latency estimate for the core phase of 12-bit LargeLUT.

Operation	Count	Per-operation/batch time (ms)	Estimated time (ms)
Blind rotation	4	53.0	212
SampleExtract + KeySwitch	624	59.9 per batch of 12	3,114
PackingKS	3	60.1	180
Total	–	–	3,507

Table 11: Performance of decomposition-based methods and Meta-WoP-PBS for arbitrary 8-to-8-bit LUT evaluation.

Method	Time*(ms)	Speedup over tree-based	L	Evaluation key size (MB)
LHE_4_4	198	2.13x	–†	49.33
Tree-based	422	1x	–†	160
Meta_8C	208	2.03x	> 290	70.74
Meta_8	266	1.59x	$> 25{,}000$	100.21

* LHE_4_4 has overall failure probability 2^{-40}. The tree-based latency is an operation-level estimate with failure probability 2^{-23}; Meta_8 and Meta_8C have overall failure probability at most 2^{-40}.

† LHE_4_4 and the tree-based method produce digit-decomposed outputs, so the single-LWE linear-combination measure L does not apply.

Springer, Cham, Switzerland, Kamakura, Japan, 460–479. doi:10.1007/978-3-030-78372-3_18

[7] Jung Hee Cheon, Wonhee Cho, Jaehyung Kim, and Damien Stehlé. 2023. Homomorphic Multiple Precision Multiplication for CKKS and Reduced Modulus Consumption. In *ACM CCS 2023: 30th Conference on Computer and Communications Security*, Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda (Eds.). ACM Press, Copenhagen, Denmark, 696–710. doi:10.1145/3576915.3623086

[8] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. 2020. TFHE: Fast Fully Homomorphic Encryption Over the Torus. *Journal of Cryptology* 33, 1 (Jan. 2020), 34–91. doi:10.1007/s00145-019-09319-x

[9] Ilaria Chillotti, Marc Joye, and Pascal Paillier. 2021. Programmable Bootstrapping Enables Efficient Homomorphic Inference of Deep Neural Networks. In *Cyber Security Cryptography and Machine Learning*, Shlomi Dolev, Oded Margalit, Benny Pinkas, and Alexander Schwarzmann (Eds.). Springer International Publishing, Cham, 1–19.

[10] Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. 2021. Improved Programmable Bootstrapping with Larger Precision and Efficient Arithmetic Circuits for TFHE. In *Advances in Cryptology – ASIACRYPT 2021, Part III (Lecture Notes in Computer Science, Vol. 13092)*, Mehdi Tibouchi and Huaxiong Wang (Eds.). Springer, Cham, Switzerland, Singapore, 670–699. doi:10.1007/978-3-030-92078-4_23

[11] Léo Ducas and Daniele Micciancio. 2015. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In *Advances in Cryptology – EUROCRYPT 2015, Part I (Lecture Notes in Computer Science, Vol. 9056)*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer Berlin Heidelberg, Germany, Sofia, Bulgaria, 617–640. doi:10.1007/978-3-662-46800-5_24

[12] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *41st Annual ACM Symposium on Theory of Computing*, Michael Mitzenmacher (Ed.). ACM Press, Bethesda, MD, USA, 169–178. doi:10.1145/1536414.1536440

[13] Antonio Guimarães, Edson Borin, and Diego F. Aranha. 2021. Revisiting the functional bootstrap in TFHE. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021, 2 (2021), 229–253. doi:10.46586/tches.v2021.i2.229-253

[14] Intak Hwang. 2023. TFHE-go. Online: https://github.com/sp301415/tfhe-go.

[15] Changmin Lee, Seonhong Min, Jinyeong Seo, and Yongsoo Song. 2023. Faster TFHE Bootstrapping with Block Binary Keys. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security* (Melbourne, VIC, Australia) *(ASIA CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2–13. doi:10.1145/3579856.3595804

[16] KangHoon Lee and Ji Won Yoon. 2023. Discretization Error Reduction for High Precision Torus Fully Homomorphic Encryption. In *PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part II (Lecture Notes in Computer Science, Vol. 13941)*, Alexandra Boldyreva and Vladimir Kolesnikov (Eds.). Springer, Cham, Switzerland, Atlanta, GA, USA, 33–62. doi:10.1007/978-3-031-31371-4_2

[17] Kang Hoon Lee and Ji Won Yoon. 2026. Homomorphic Field Trace Revisited: Breaking the Cubic Noise Barrier. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2026, 1 (2026), 105–134.

[18] Zeyu Liu, Daniele Micciancio, and Yuriy Polyakov. 2022. Large-Precision Homomorphic Sign Evaluation Using FHEW/TFHE Bootstrapping. In *Advances in Cryptology – ASIACRYPT 2022, Part II (Lecture Notes in Computer Science, Vol. 13792)*, Shweta Agrawal and Dongdai Lin (Eds.). Springer, Cham, Switzerland, Taipei, Taiwan, 130–160. doi:10.1007/978-3-031-22966-4_5

[19] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. 2010. On Ideal Lattices and Learning with Errors over Rings. In *Advances in Cryptology – EUROCRYPT 2010 (Lecture Notes in Computer Science, Vol. 6110)*, Henri Gilbert (Ed.). Springer Berlin Heidelberg, Germany, French Riviera, 1–23. doi:10.1007/978-3-642-13190-5_1

[20] Shihe Ma, Tairong Huang, Anyu Wang, Qixian Zhou, and Xiaoyun Wang. 2024. Fast and Accurate: Efficient Full-Domain Functional Bootstrap and Digit Decomposition for Homomorphic Computation. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2024, 1 (2024), 592–616. doi:10.46586/tches.v2024.i1.592-616

[21] Takumi Nishimura, Kazunari Tozawa, and Kunihiko Sadakane. 2025. Surpassing the Word Size Limitation of TFHE with Noise Calibration. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (Taipei, Taiwan) *(CCS '25)*. Association for Computing Machinery, New York, NY, USA, 438–452. doi:10.1145/3719027.3744808

[22] Oded Regev. 2009. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* 56, 6, Article 34 (Sept. 2009), 40 pages. doi:10.1145/1568318.1568324

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

[24] Daphné Trama, Aymen Boudguiga, Pierre-Emmanuel Clet, Renaud Sirdey, and Nicolas Ye. 2025. Designing a General-Purpose 8-bit (T)FHE Processor Abstraction. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2025, 2 (Mar. 2025), 535–578. doi:10.46586/tches.v2025.i2.535-578

[25] Ruida Wang, Jincheol Ha, Xuan Shen, Xianhui Lu, Chunling Chen, Kunpeng Wang, and Jooyoung Lee. 2025. Refined TFHE Leveled Homomorphic Evaluation and Its Application. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (Taipei, Taiwan) *(CCS '25)*. Association for Computing Machinery, New York, NY, USA, 2309–2323. doi:10.1145/3719027.3744873

[26] Ruida Wang, Yundi Wen, Zhihao Li, Xianhui Lu, Benqiang Wei, Kun Liu, and Kunpeng Wang. 2024. Circuit Bootstrapping: Faster and Smaller. In *Advances in Cryptology – EUROCRYPT 2024, Part II (Lecture Notes in Computer Science, Vol. 14652)*, Marc Joye and Gregor Leander (Eds.). Springer, Cham, Switzerland, Zurich, Switzerland, 342–372. doi:10.1007/978-3-031-58723-8_12

[27] Zama. 2022. TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data. https://github.com/zama-ai/tfhe-rs.

A Generative AI Usage

We used ChatGPT to improve the presentation of the introduction, suggest figure content, proofread the manuscript, and help inspect the implementation. The authors manually verified all resulting text, proofs, and code corrections. The first versions of the manuscript and implementation were written without AI, and no citations were generated using AI.

B Open Science

The implementation of Meta-PBS is available at https://github.com/msh086/META-FBS. It can be used to reproduce the benchmark results in Section 6. More details on its usage can be found in the README file.

C More Details on Meta-WoP-PBS

The full algorithm of Meta-WoP-PBS is presented in Algorithm 3. The initial redundancy is at most $2^{-\nu}\frac{N}{t}$ instead of $2^{-\nu}\frac{2N}{t}$ in Meta-PBSmanyLUTs because the actual plaintext modulus is doubled to $2t$.

Corollary 1. *Let V_{ks} and V_{ms} denote the noise variances introduced by key switching and modulus switching, respectively. Assume all the conditions in Theorem 5 hold, except that the bounds on r_0 and δ_i are replaced by $r_0 \le \frac{N\cdot 2^{-\nu}}{t}$ and*

$$\delta_i \ge \left\lceil \mathrm{erfc}^{-1}(p_{\mathrm{fail}}) \cdot \sqrt{\frac{n+4}{12} + 2\left(\frac{N\prod_{j\in[i]}\beta_j}{q}\right)^2 \cdot V_{\mathrm{in}}} + \frac{1}{2} \right\rceil.$$

If, in addition,

$$\delta_{\mathrm{CS}} \ge \left\lceil \mathrm{erfc}^{-1}(p_{\mathrm{fail}}) \cdot \sqrt{2} \cdot \sqrt{\left(\frac{2N}{q}\right)^2 (V_{\mathrm{Meta-PBS}} + V_{\mathrm{ks}}) + V_{\mathrm{ms}}} + \frac{1}{2} \right\rceil,$$

$$2\delta_{\mathrm{CS}} + 1 \le (r_K - 2\delta_K)\beta_{\mathrm{CS}}, \qquad \text{and} \qquad (r_K - 2\delta_K + \epsilon_{\mathrm{CS}})\beta_{\mathrm{CS}} \le \lfloor \frac{N}{\tau} \rfloor.$$

then Algorithm 3 outputs $\{\mathrm{LWE}_{\vec{S}}(\frac{q}{t} f_k(m))\}_{k\in[2^\nu-1]}$ with failure probability at most $(K+2)\cdot p_{\mathrm{fail}}$.

Proof. The computation through C_K performs the multi-output Meta-PBS procedure without sample extraction, using 2^ν negacyclic functions $f_k^* : \mathbb{Z}_{2t} \to \mathbb{Z}_t$. Its lifted input ciphertext is

$$\widetilde{c} = [c]_{2q} = \mathrm{LWE}_{\vec{s},2q}\left(\frac{q}{t}m + q\gamma; e\right).$$

The initial HomDivRem has input modulus $2q$ and target quotient modulus $2N$. Hence, the remainder modulus in round i is

$$\frac{q}{N\prod_{j\in[i]}\beta_j}.$$

By Lemma 4, this gives the stated bound on δ_i. For $x \in [t]$, $f_{2^\nu-1}^*(x) = -t/4$ and $f_k^*(x) = f_k(x) \in \mathbb{Z}_t$ for $k \in [2^\nu-1]$. The Meta-PBSmanyLUT invariant gives that $M_K X^{e_K} X^{-kD_K}$ is r_K-redundant in $\frac{q}{t} f_k^*(m + \gamma t) = \frac{q}{t} f_k(m)(-1)^\gamma$. When $|e_K| \le \delta_K$, omitting the restoring factor X^{e_K} shortens the redundancy interval by at most δ_K on each side, so $\mathrm{Dec}(C_K)X^{-kD_K}$ is $(r_K - 2\delta_K)$-redundant in the same value.

Algorithm 3 Meta-WoP-PBS

Require: Meta-PBS parameters from Table 1
Require: Meta-PBSmanyLUT parameter ν and $D_i = r_0 \prod_{j\in[i]}\beta_j$ for $i \in [K+1]$
Require: Parameter τ, the number of messages processed in a single CancelSign
Require: Perturbation bound δ_{CS} and HomTruncRepeat* parameters $\beta_{\mathrm{CS}}, \epsilon_{\mathrm{CS}}$ for CancelSign
Require: $2^\nu - 1$ arbitrary functions $\{f_k\}_{k\in[2^\nu-1]}$ over $\mathbb{Z}_t$
Require: LWE ciphertext $c = \mathrm{LWE}_{\vec{s},q}(\frac{q}{t}m; e)$ with $\mathrm{Var}(e) \le V_{\mathrm{in}}$
Require: Blind rotation keys $\mathrm{BRK}^{(B_{\mathrm{br}},l_{\mathrm{br}})}$
Ensure: $c_{\mathrm{out},k} = \mathrm{LWE}_{\vec{S}}(\frac{q}{t} f_k(m))$ for $k \in [2^\nu - 1]$

Define $f_{2^\nu-1} : \mathbb{Z}_t \to \mathbb{Z}_t$ as the constant function $f_{2^\nu-1}(x) = -t/4$
Lift c component-wise to $\widetilde{c} = [c]_{2q} = \mathrm{LWE}_{\vec{s},2q}(\frac{q}{t}m + q\gamma; e)$, where $\gamma \in \{0,1\}$
$\mathrm{TV} \leftarrow \sum_{i\in[t]}\sum_{k\in[2^\nu]} X^{i\cdot N/t + k\cdot r_0}\frac{q}{t} f_k(i) \cdot \sum_{j\in[r_0]_{\mathrm{sym}}} X^j$
$c_0, c_{\mathrm{rem},0} \leftarrow \mathrm{HomDivRem}(\widetilde{c}, 2N)$
$C_0 \leftarrow \mathrm{BlindRotate}(c_0, \mathrm{TV}, \mathrm{BRK}^{(B_{\mathrm{br}},l_{\mathrm{br}})})$
for $i \in [K]$ **do**
  $C_i' \leftarrow \mathrm{HomTruncRepeat}_{r_i}(C_i, [\![-T_i, T_i + D_i\cdot(2^\nu-1)]\!], \beta_i)$
  $c_{i+1}, c_{\mathrm{rem},i+1} \leftarrow \mathrm{HomDivRem}(c_{\mathrm{rem},i}, \beta_i)$
  $C_{i+1} \leftarrow \mathrm{BlindRotate}(c_{i+1}, C_i', \mathrm{BRK}^{(B_{\mathrm{br}},l_{\mathrm{br}})})$
end for
$c_{\mathrm{sgn}} \leftarrow \mathrm{ModSwitch}(\mathrm{KeySwitch}(\mathrm{SampleExtract}(C_K, D_K(2^\nu - 1)), \mathrm{KSK}_{\vec{S}\to\vec{s}}), 2N) + \frac{N}{2}$
for $j \in [\lceil \frac{2^\nu-1}{\tau} \rceil]$ **do**
  $C^* \leftarrow \sum_{k\in[\![j\tau, \min((j+1)\tau, 2^\nu-2)]\!]} \mathrm{HomTruncRepeat}^*_{r_K-2\delta_K}(C_K \cdot X^{-kD_K}, [r_K - 2\delta_K]_{\mathrm{sym}}, \beta_{\mathrm{CS}}, \epsilon_{\mathrm{CS}}) \cdot X^{(k-j\tau)\lfloor N/\tau \rfloor}$
  $C_{\mathrm{out}} \leftarrow \mathrm{BlindRotate}(c_{\mathrm{sgn}}, C^*, \mathrm{BRK}^{(B_{\mathrm{br}},l_{\mathrm{br}})})$
  for $k \in [\![j\tau, \min((j+1)\tau, 2^\nu - 2)]\!]$ **do**
    $c_{\mathrm{out},k} \leftarrow \mathrm{SampleExtract}(C_{\mathrm{out}}, (k - j\tau)\cdot\lfloor N/\tau \rfloor)$
  end for
end for

The ciphertext c_{sgn} encrypts

$$\frac{2N}{t} f_{2^\nu-1}^*(m + \gamma t) + \frac{N}{2} = \frac{N}{2}(-1)^{\gamma+1} + \frac{N}{2} = N\gamma.$$

Denote the LWE noise in c_{sgn} by e_{sgn}. Its variance is

$$\mathrm{Var}(e_{\mathrm{sgn}}) = \left(\frac{2N}{q}\right)^2 (V_{\mathrm{Meta-PBS}} + V_{\mathrm{ks}}) + V_{\mathrm{ms}}.$$

Thus, $\Pr[|e_{\mathrm{sgn}}| > \delta_{\mathrm{CS}}] < p_{\mathrm{fail}}$.

In the CancelSign stage, $(r_K - 2\delta_K + \epsilon_{\mathrm{CS}})\beta_{\mathrm{CS}} \le \lfloor N/\tau \rfloor$ ensures that the supports of different HomTruncRepeat* outputs do not overlap. $\mathrm{Dec}(C^*)\cdot X^{-(k-j\tau)\cdot\lfloor N/\tau\rfloor}$ is then $(r_K - 2\delta_K)\beta_{\mathrm{CS}}$-redundant in $\frac{q}{t} f_k(m)\cdot(-1)^\gamma$. Since $\mathrm{Dec}(C_{\mathrm{out}}) = \mathrm{Dec}(C^*)\cdot X^{-N\gamma - e_{\mathrm{sgn}}}$, $\mathrm{Dec}(C_{\mathrm{out}})\cdot X^{-(k-j\tau)\cdot\lfloor N/\tau\rfloor}$ is $((r_K - 2\delta_K)\beta_{\mathrm{CS}} - 2\delta_{\mathrm{CS}})$-redundant in $\frac{q}{t} f_k(m)$.

Finally, the correctness of SampleExtract is guaranteed by $(r_K - 2\delta_K)\beta_{\mathrm{CS}} - 2\delta_{\mathrm{CS}} \ge 1$. The required high-probability events bound $e_0, \ldots, e_K$ and e_{sgn}, for $K+2$ events in total; the union bound gives the stated overall failure probability.

□

Corollary 2. *Let $\tau_j = |[\![j\tau, \min((j+1)\tau, 2^\nu - 2)]\!]|$ be the actual number of messages in batch j. Each ciphertext output for that batch has noise variance*

$$V_{\text{Meta-WoP-PBS}} = V_{\text{Meta-PBS}} + V_{\text{br}} + V_{\text{fill}}(\epsilon_{\text{CS}}) + (\tau_j - 1)V_{\text{emp}}(\epsilon_{\text{CS}}).$$

Proof. The useful coefficients in C_{out} have the following noise components.

- The noise from C_K, with a variance of $V_{\text{Meta-PBS}}$.
- The noise from the HomTruncRepeat* that sets this coefficient, with variance $V_{\text{fill}}(\epsilon_{\text{CS}})$.
- The noise from the other $\tau_j - 1$ HomTruncRepeat* calls in the batch, with total variance $(\tau_j - 1)V_{\text{emp}}(\epsilon_{\text{CS}})$.

The final blind rotation adds a variance of V_{br}. □

D Proofs

We need the following lemma to prove Lemma 5.

Lemma 7. *For integers $-N < a \le 0 \le b < N$, $\beta > 0$ and i such that $-N < a - i \le b - i < N$,*

$$\text{TruncPad}(M \cdot X^i, [\![a, b]\!], \beta) = \text{TruncPad}(M, [\![a - i, b - i]\!], \beta) \cdot X^{i\beta}.$$

Proof. For $i \ge 0$,

$$\begin{aligned}
\text{TruncPad}(M \cdot X^i, [\![a, b]\!], \beta) &= \sum_{j \in [\![0,b]\!]} (M \cdot X^i)[j] \cdot X^{j\beta} + \sum_{j \in [\![a,-1]\!]} (M \cdot X^i)[N + j] \cdot X^{N+j\beta} \\
&= -\sum_{j \in [\![0, \min(i-1, b)]\!]} M[N + j - i] \cdot X^{j\beta} + \sum_{j \in [\![\min(i, b+1), b]\!]} M[j - i] \cdot X^{j\beta} + \sum_{j \in [\![a,-1]\!]} M[N + j - i] \cdot X^{N+j\beta} \\
&= X^{i\beta} \cdot \left(-\sum_{k \in [\![-i, \min(-1, b-i)]\!]} M[N + k] \cdot X^{k\beta} + \sum_{k \in [\![\min(0, b-i+1), b-i]\!]} M[k] \cdot X^{k\beta} + \sum_{k \in [\![a-i,-i-1]\!]} M[N + k] \cdot X^{N+k\beta} \right) \\
&= X^{i\beta} \cdot \text{TruncPad}(M, [\![a - i, b - i]\!], \beta).
\end{aligned}$$

The case of $i < 0$ can be proved similarly. □

Lemma 5. For brevity, let

$$H_i^* = \text{HomTruncPad}^*(\cdot, [\![-T_i, T_i]\!], \beta_i, \epsilon_i), \qquad H_i = \text{HomTruncPad}(\cdot, [\![-T_i, T_i]\!], \beta_i).$$

Replacing HomTruncRepeat by HomTruncRepeat* preserves the correctness of Meta-PBS provided that the optimized operation keeps all useful coefficients of the original operation. Since both operations duplicate each HomTruncPad output coefficient β_i times, it suffices to prove that H_i^* agrees with H_i on the latter's nonzero coefficients, and that every extra coefficient of H_i^* is at least β_i positions away from them.

In other words, we aim to prove that the coefficient of the X^i term in $\text{Dec}(H_i^* - H_i)$ is zero for $0 \le i < (T_i + 1)\beta_i$ and $N - (T_i + 1)\beta_i < i < N$.

Recall that $\text{HomTruncPad}^*((A, B) = \text{RLWE}(M), [\![a, b]\!], \beta, \epsilon)$ decrypts to

$$\sum_{j \in [\lceil N/(\epsilon+1) \rceil]} \text{TruncPad}(A \cdot X^{j(\epsilon+1)}, [\![a - \epsilon, b]\!], \beta) \cdot \sum_{i \in [\![j(\epsilon+1), \min(j(\epsilon+1)+\epsilon, N-1)]\!]} S[i] \cdot X^{\beta(i - j(\epsilon+1))} + \text{TruncPad}(B, [\![a, b]\!], \beta),$$

while $\text{HomTruncPad}((A, B) = \text{RLWE}(M), [\![a, b]\!], \beta)$ decrypts to

$$\sum_{j \in [N]} \text{TruncPad}(A \cdot X^j, [\![a, b]\!], \beta) \cdot S[j] + \text{TruncPad}(B, [\![a, b]\!], \beta).$$

Taking the difference of the two equations, we observe that the difference is the sum of the following expression with $k = 0, \epsilon + 1, 2(\epsilon + 1), \cdots$.

$$\begin{aligned}
&\text{TruncPad}(A \cdot X^k, [\![a - \epsilon, b]\!], \beta) \cdot \sum_{i \in [\![k, \min(k+\epsilon, N-1)]\!]} S[i] \cdot X^{\beta(i-k)} - \sum_{i \in [\![k, \min(k+\epsilon, N-1)]\!]} \text{TruncPad}(A \cdot X^i, [\![a, b]\!], \beta) \cdot S[i] \\
&= \sum_{i \in [\![k, \min(k+\epsilon, N-1)]\!]} S[i] \cdot \left(\text{TruncPad}(A \cdot X^k, [\![a - \epsilon, b]\!], \beta) \cdot X^{\beta(i-k)} - \text{TruncPad}(A \cdot X^i, [\![a, b]\!], \beta) \right) \\
&= \sum_{i \in [\![k, \min(k+\epsilon, N-1)]\!]} S[i] \cdot \left(\text{TruncPad}(A \cdot X^i, [\![a - \epsilon + i - k, b + i - k]\!], \beta) - \text{TruncPad}(A \cdot X^i, [\![a, b]\!], \beta) \right) \\
&= \sum_{i \in [\![k, \min(k+\epsilon, N-1)]\!]} S[i] \cdot \left(\sum_{j \in [\![b+1, b+i-k]\!]} (A \cdot X^i)[j] \cdot X^{j\beta} + \sum_{j \in [\![a-\epsilon+i-k, a-1]\!]} (A \cdot X^i)[N + j] \cdot X^{N+j\beta} \right),
\end{aligned}$$

where the second equality follows from Lemma 7.

According to the indexing of j, if we want the difference polynomial to have only zero coefficients for degrees in $[\![0, (b+1)\beta - 1]\!] \bigcup [\![N + (a-1)\beta + 1, N - 1]\!]$, we need $(b + \epsilon)\beta \le N + (a-1)\beta$ and $N + (a - \epsilon - 1)\beta \ge b\beta$. Substituting $a = -T_i$, $b = T_i$, $\beta = \beta_i$, and $\epsilon = \epsilon_i$ makes both inequalities equivalent to $(2T_i + 1 + \epsilon_i)\beta_i \le N$, so the modified Condition 3 preserves every coefficient needed by the correctness proof.

□

Lemma 6. Following the analysis of PackingKS in [10], $E'[i]$ for useful coefficients consists of the following parts for $i \in [\mathrm{supp}(\mathrm{TruncRepeat}_r(M, [\![a,b]\!], \beta))]_N$.

The first component is equal to one of the coefficients of E according to the definition of TruncRepeat, and therefore has variance σ^2.

The second component is the inner product between the coefficients of S and a vector of decomposition residues. As in PackingKS, its variance is $\frac{N}{2}\left(\frac{q^2}{12B_{\mathrm{tr}}^{2l_{\mathrm{tr}}}} - \frac{1}{12}\right) + \frac{N}{16}$.

The third component comes from the encryption noise in RLev ciphertexts. When $(\epsilon+1) \mid N$, $G^{(B_{\mathrm{tr}},l_{\mathrm{tr}},q)}$ acts on $\frac{N}{\epsilon+1}$ polynomials, each with a support size of $b-a+1+\epsilon$. The final noise polynomial is the inner product between $\frac{N}{\epsilon+1}l_{\mathrm{tr}}$ decomposed polynomials and a vector of encryption noises in RLev ciphertexts, and the final component is a coefficient in this polynomial. Thus, it has a variance of $(b-a+1+\epsilon)\frac{N}{\epsilon+1}l_{\mathrm{tr}}\frac{B_{\mathrm{tr}}^2+2}{12}\sigma_{\mathrm{trk}}^2 = \left((b-a)\frac{N}{\epsilon+1}+N\right)l_{\mathrm{tr}}\frac{B_{\mathrm{tr}}^2+2}{12}\sigma_{\mathrm{trk}}^2$.

Now let $\Gamma = \lceil \frac{N}{\epsilon+1} \rceil$ be the number of polynomials to be gadget decomposed. Suppose $\frac{N}{\Gamma} \le \epsilon+1 < \frac{N}{\Gamma-1}$, there are $\Gamma-1$ polynomials with support size $b-a+1+\epsilon$ and a remaining one with support size $b-a+1+N-(\Gamma-1)(\epsilon+1)-1$. Thus, the variance is $((b-a+1+\epsilon)(\Gamma-1)+b-a+1+N-(\Gamma-1)(\epsilon+1)-1)l_{\mathrm{tr}}\frac{B_{\mathrm{tr}}^2+2}{12}\sigma_{\mathrm{trk}}^2 = ((b-a)\Gamma+N)l_{\mathrm{tr}}\frac{B_{\mathrm{tr}}^2+2}{12}\sigma_{\mathrm{trk}}^2$.

We also need to take the FFT noise into account, which, following the heuristic formula for FFT noise variance in blind rotation [3], is estimated as $((b-a)\Gamma+N)l_{\mathrm{tr}}B_{\mathrm{tr}}^2 N \cdot 2^{19.4}$.

The conclusion on V_{emp} holds because the third component (with FFT noise) is the only noise component in that case.

□

E Comparing HomTruncRepeat* with Other Packing Methods

Table 12 summarizes the performance of multiple packing methods to compute $\mathrm{TruncRepeat}_{r_i}(\cdot, [\![-T_i, T_i]\!], \beta_i)$ homomorphically. Note that although HomTruncRepeat* is more efficient for this special task, it introduces garbage coefficients, which may be undesirable for other packing tasks.

We now describe how to obtain the time cost in the last row. With a packing base of one, the automorphism-based packing is given N LWE ciphertexts as inputs and uses a depth-$\log N+1$ binary tree to combine them into a single RLWE ciphertext. With the leaf node layer as the zeroth layer, the i-th layer includes $N/2^i$ RLWE ciphertexts, where the j-th one contains the input LWE messages with indices in $j + N/2^i \cdot [2^i]$. Each non-leaf node is obtained using a homomorphic automorphism, costing a total of $N-1$ ones.

To compute TruncRepeat, only the $2T_i+1$ input coefficients around the constant term are retained, each repeated β_i times. After shifting these inputs into the first $\beta_i(2T_i+1)$ leaf nodes, the remaining leaves are zero ciphertexts and their subtrees can be pruned; the effect of the shift can be undone after packing. In layer i, there are $\max(0, N/2^i - \beta_i(2T_i+1))$ zero nodes. Since the full tree has $N-1$ non-leaf nodes, pruning leaves

$$(N-1) - \sum_{i=1}^{\log N-\kappa} (N/2^i - \beta_i(2T_i+1)) = 2^\kappa - 1 + (\log N - \kappa)\beta_i(2T_i+1)$$

nonzero non-leaf nodes. The parameters used in Section 6 satisfy $2^\kappa \ge N/2$; the packing step therefore requires at least $3N/4$ automorphisms.

F Automatic Parameter Selection

Let $\lambda = 2$ for Meta-PBS and Meta-PBSmanyLUT, and let $\lambda = 1$ for Meta-WoP-PBS, whose lifted input encodes a function over $\mathbb{Z}_{2t}$. For $i \in [K+1]$, define

$$\alpha_i = \frac{\lambda N}{t}\prod_{j\in[i]}\beta_j.$$

The initial-redundancy condition and the redundancy-increase chain imply $2^\nu \max(r_i, D_i) \le \alpha_i$.

Write $\rho = \mathrm{erfc}^{-1}(p_{\mathrm{fail}})$. We measure the input noise by $c_{\mathrm{meta}} = \log_2(q/(2t)) - \log_2(\rho\sqrt{2V_{\mathrm{in}}})$. For a fixed choice of $\boldsymbol{\beta}$, the perturbation bound in Condition 4 of Theorem 5, including its Meta-WoP-PBS modification in Corollary 1, is

$$\Delta_i(\alpha_i) := \left\lceil \rho\sqrt{\frac{n+4}{12} + 2\left(\frac{\alpha_i t}{q}\right)^2 V_{\mathrm{in}}} + \frac{1}{2} \right\rceil.$$

Thus, a valid parameter set must take $\delta_i \ge \Delta_i(\alpha_i)$. The search uses equality unless an additional safety margin is requested.

The automatic parameter-selection procedure consists of the following stages.

(1) *Set the capacity and final redundancy targets.*
The subroutine receives lower-bound targets c^*_{meta} and c^*_{Hom} for the input and output noise capacities. It also receives an integer target r^*_{ext} for the residual redundancy after the final iteration $r_K - 2\delta_K$, and the batch size τ of CancelSign. For ordinary Meta-PBS and Meta-PBSmanyLUT, we set $r^*_{\mathrm{ext}} = 1$, which is exactly the final-redundancy requirement. For Meta-WoP-PBS, larger values may reduce the cost of the subsequent CancelSign stage and are therefore included in the search space.

(2) *Estimate a heuristic iteration count $\widehat{K}$.*
The final redundancy target requires

$$r_{\widehat{K}} \ge 2\Delta_{\widehat{K}}(\alpha_{\widehat{K}}) + r^*_{\mathrm{ext}}.$$

Before $\boldsymbol{\beta}$ is known, replacing $\alpha_{\widehat{K}}$ by its lower bound $2^\nu r_{\widehat{K}}$ and dropping the ceiling gives the necessary but insufficient inequality

$$r_{\widehat{K}} \ge 2\rho\sqrt{\frac{n+4}{12} + 2\left(\frac{2^\nu r_{\widehat{K}} t}{q}\right)^2 V_{\mathrm{in}}} + 1 + r^*_{\mathrm{ext}}.$$

Let $h = 1 + r^*_{\mathrm{ext}}$ and $C = (n+4)/12$. Subject to $r_{\widehat{K}} \ge h$, squaring the inequality yields $a r_{\widehat{K}}^2 + b r_{\widehat{K}} + c \ge 0$, where

$$a = 1 - 8\rho^2 2^{2\nu}\left(\frac{t}{q}\right)^2 V_{\mathrm{in}},$$
$$b = -2h,$$
$$c = h^2 - 4\rho^2 C.$$

In the usual parameter regime $2^\nu 2^{-c^*_{\mathrm{meta}}} < 1$, we have $a > 0$, and the larger root gives an optimistic lower bound on $r_{\widehat{K}}$. The exact integer constraint above is checked after $\boldsymbol{\beta}$ has been fixed.

Table 12: The computational cost, evaluation key size, and variance of introduced noise of different packing methods. Time cost is measured in the number of integer multiplications, assuming each FFT in R costs $cN\log N$. Key size is measured in the number of polynomial coefficients.

Packing method	Time cost†	Key size*	Noise variance (w/o. FFT noise)
HomTruncRepeat*	$c\lceil\frac{N}{\epsilon+1}\rceil lN\log N$	$2\lceil\frac{N}{\epsilon+1}\rceil lN$	$(2T_i\lceil\frac{N}{\epsilon+1}\rceil+N)\frac{lB^2\sigma^2}{12}+\frac{Nq^2}{24B^{2l}}$
PubFuncKS	$clN^2\log N$	$2lN^2$	$\beta_i(2T_i+1)N\frac{lB^2\sigma^2}{12}+\frac{Nq^2}{24B^{2l}}$
Base-aware PackingKS (scalar mult.)	$(2T_i+1)lN^2$	$2lN^2$	$(2T_i+1)N\frac{lB^2\sigma^2}{12}+\frac{Nq^2}{24B^{2l}}$
Base-aware PackingKS (polynomial mult.)	$clN^2\log N$	$2lN^2$	$(2T_i+1)N\frac{lB^2\sigma^2}{12}+\frac{Nq^2}{24B^{2l}}$
Automorphism-based packing (power-of-two β_i)	$c(2T_i+1+\log(N/\beta_i))lN\log N$	$2lN\log N$	$\beta_i\log N(N\frac{lB^2\sigma^2}{12}+\frac{Nq^2}{24B^{2l}})$
Automorphism-based packing (non-power-of-two β_i)	$c(2^{\kappa}-1+(\log N-\kappa)\beta_i(2T_i+1))lN\log N$	$2lN\log N$	$\log N(N\frac{lB^2\sigma^2}{12}+\frac{Nq^2}{24B^{2l}})$

* The evaluation keys can be shared across iterations for (a) HomTruncRepeat* with the same β_i and ϵ_i, (b) base-aware PackingKS with the same β_i, and (c) PubFuncKS and automorphism-based packing under arbitrary parameters. The centering monomial $X^{\Delta_{r_i,\beta_i}}$ can be applied separately and need not be embedded in the keys.

† κ in the last row is the smallest integer satisfying $2^{\kappa}\geq\beta_i(2T_i+1)$.

Starting from this estimate of $r_{\widehat{K}}$, the search obtains an optimistic upper bound on each β_i in reverse order. For a candidate β_i, define

$$r_i^{\mathrm{lb}}=\left\lceil\frac{r_{i+1}}{\beta_i}\right\rceil,\qquad D_i^{\mathrm{lb}}=r_i^{\mathrm{lb}},$$
$$\alpha_i^{\mathrm{lb}}=2^{\nu}r_i^{\mathrm{lb}},\qquad \delta_i^{\mathrm{lb}}=\Delta_i(\alpha_i^{\mathrm{lb}}),$$
$$T_i^{\mathrm{lb}}=\delta_i^{\mathrm{lb}}+\left\lfloor\frac{r_i^{\mathrm{lb}}}{2}\right\rfloor.$$

Since $D_i\geq r_i$ and $\epsilon_i\geq 0$, every valid parameter set must satisfy the relaxed support constraint

$$\beta_i\left(2T_i^{\mathrm{lb}}+1+(2^{\nu}-1)D_i^{\mathrm{lb}}\right)\leq N.$$

The largest β_i passing this test is used only to estimate $\widehat{K}$; further enumeration is required to find valid parameters.

(3) *Enumerate and verify the structural parameters.*
Let $\widehat{K}$ denote the heuristic iteration-count estimate obtained above, and set $K_{\max}=\widehat{K}+1$ as the maximum recursion depth. Neither $\widehat{K}$ nor $K_{\max}$ fixes the iteration count of an accepted candidate.
Starting from the target final redundancy, the recursion selects packing bases in reverse order and tracks the redundancy required before the selected sequence. At each level, it computes an optimistic upper bound on the next packing base as in the preceding stage and explores every admissible value within that bound.
If the required redundancy does not exceed $r_0^{\max}=2^{-\nu}\alpha_0$, the branch is accepted. The number of selected bases defines its actual iteration count K, and the bases are then indexed in forward execution order as $\boldsymbol{\beta}=(\beta_0,\ldots,\beta_{K-1})$. If the required redundancy still exceeds $r_0^{\max}$ after $K_{\max}$ bases have been selected, the branch is rejected as infeasible.

For each accepted $\boldsymbol{\beta}$, the search computes the exact α_i and $\delta_i=\Delta_i(\alpha_i)$, sets

$$r_K=2\delta_K+r_{\mathrm{ext}}^{*},\qquad r_i=\left\lceil\frac{r_{i+1}}{\beta_i}\right\rceil,$$

and derives

$$D_i=r_0\prod_{j\in[i]}\beta_j,\qquad T_i=\delta_i+\left\lfloor\frac{r_i}{2}\right\rfloor.$$

For the optimized construction, set the column-merging parameter to its largest admissible value:

$$\epsilon_i=\left\lfloor\frac{N}{\beta_i}\right\rfloor-(2T_i+1+(2^{\nu}-1)D_i).$$

Candidates with $\epsilon_i<0$ are discarded. All remaining candidates are checked against Conditions 1–4 of Theorem 5 and the modified support constraint for HomTruncRepeat*. Meta-WoP-PBS candidates are additionally checked against the modified initial-redundancy and perturbation bounds and all CancelSign constraints in Corollary 1.

(4) *Choose the gadget parameters.*
For every valid structural candidate, the search enumerates the gadget lengths $(l_{\mathrm{br}},l_{\mathrm{tr}},l_{\mathrm{ks}})$. For each length, it finds the zero-derivative point of the corresponding variance expression and chooses the lower-variance of the two neighboring integer bases. For the HomTruncRepeat parameters in Meta-WoP-PBS, this stage first uses the minimum admissible value of β_{CS} that guarantees correct sample extraction and then uses the maximum admissible value of ϵ_{CS}.
A completed candidate is retained only if it provides $c_{\mathrm{meta}}\geq c_{\mathrm{meta}}^{*}$ and $c_{\mathrm{Hom}}\geq c_{\mathrm{Hom}}^{*}$ and satisfies every applicable correctness constraint. Its computational cost is estimated by the number of required FFTs and inverse FFTs. The candidate with the smallest estimated cost is selected, with lexicographic order used only to break ties.

The top-level search is summarized in Algorithm 4. Because both the noise model and the FFT-based cost model are approximate and the candidate ranges are finite, the result is a heuristic optimum within the configured search space, not a proof of global optimality.

Algorithm 4 Automatic Parameter Selection

Require: Basic PBS parameters N, n, q, t, ν and the target variant
Require: Per-event failure probability p_{fail}
Require: CancelSign batch size τ when the target variant is Meta-WoP-PBS
Require: Target output capacity c^*_{Hom}
Require: Candidate target sets S_c for c^*_{meta} and S_r for r^*_{ext}
Ensure: The feasible parameter set with the smallest estimated cost in the configured search space

BestParams $\leftarrow \perp$, MinCost $\leftarrow +\infty$
for all $(c^*_{\text{meta}}, r^*_{\text{ext}}) \in S_c \times S_r$ **do**
  $\mathcal{P} \leftarrow \text{EnumerateStructure}(c^*_{\text{meta}}, r^*_{\text{ext}}, p_{\text{fail}})$
  for all $P \in \mathcal{P}$ **do**
    $\widehat{P} \leftarrow \text{CompleteAndValidate}(P, c^*_{\text{Hom}}, p_{\text{fail}}, \tau)$
    if $\widehat{P} \neq \perp$ **then**
      Cost $\leftarrow \text{EstimateFFTCount}(\widehat{P})$
      if Cost < MinCost **then**
        BestParams $\leftarrow \widehat{P}$
        MinCost $\leftarrow$ Cost
      end if
    end if
  end for
end for
return BestParams

G More details on experiments

G.1 Comparing with EBS

The full parameter sets for EBS and Meta-PBS are displayed in Table 13.

G.2 Comparing with integer-input LHE

We did not implement integer-input LHE in TFHE-go, but used the existing implementation based on TFHE-rs[6] instead. A potential problem is that the running time of the same operation may vary in different libraries, affecting the fairness of comparison. We address this issue by benchmarking the Extr.+Refr. stages of integer-input LHE (i.e., four 'blind rotation followed by key switching') in TFHE-go, while the running time of the Conv. stage is still measured in TFHE-rs. For the Extr.+Refr. part in LHE_8, its implementation in TFHE-go and TFHE-rs takes 4×74.4 ms and 4×64.7 ms, respectively. The timing of this part is measured in TFHE-go to ensure a fair comparison.

The full parameter sets for Int-LHE and Meta-PBS are displayed in Table 14.

H Output Noise Distribution

Although the blind rotation error has been verified to be a centered Gaussian [8], the design of Meta-PBS brings several subtleties to its noise analysis. Specifically, multiple blind rotations are performed on the same RLWE ciphertext with the same blind rotation keys, and the coefficients of noise terms are duplicated by TruncRepeat, which may introduce dependency between different contributing noise components and/or coefficients of the output noise. This stresses the need to validate our analysis on output noise distribution.

This raises two questions: (1) if each coefficient in the output noise is a centered Gaussian with the expected standard deviation, and (2) if the coefficients in the output noise are independent.

For the coefficient-wise distribution, our experiments show that the output noise is a biased Gaussian with lower-than-estimated variance; both the bias and variance depend strongly on the FHE library. We propose a heuristic and efficient fix to remove the bias in the output noise by randomizing the CMux gates in blind rotation.

For noise independence, we argue that the independence between noise coefficients follows from commonly used independence assumptions.

H.1 Distribution of Each Noise Coefficient

Before starting, we note that tfhe-go implements $\text{CMux}(\text{RGSW}(s_i), \text{ACC}, \text{ACC} \cdot X^{-a_i})$ as

$$\text{ACC} + (\text{ACC} \boxdot \text{RGSW}(s_i)) \cdot (X^{-a_i} - 1),$$

instead of the standard way of

$$(\text{ACC} \cdot (X^{-a_i} - 1)) \boxdot \text{RGSW}(s_i),$$

as implemented in tfhe-rs. Performing the multiplication by $X^{-a_i}-1$ after the external product $\boxdot$ has the side effect of doubling the external-product noise variance.

However, after fixing this issue, we find that the output noise distribution of Meta-PBS is close to a Gaussian distribution with a nonzero mean and a below-estimation standard deviation (see Figure 3b). To investigate whether this behavior is library-dependent, we perform three consecutive blind rotations using LWE ciphertexts with moduli of $2N$, 5, and 4 in order in both tfhe-go and tfhe-rs. One can think of the computation as a 9-bit Meta-PBS without the TruncRepeat part. It turns out that output noise bias in tfhe-rs is much smaller, but still non-negligible compared to the acceptable bias-to-stddev ratio of 0.3% (see Figure 3c and Figure 3d). Moreover, in tfhe-rs, the noise bias is still non-negligible even for a single blind rotation using an LWE ciphertext modulo $2N$, where the bias-to-stddev ratio is 1.37%. Additionally, the measured standard deviation in tfhe-rs is smaller than that in tfhe-go.

An interesting finding is that the noise bias of Meta-PBS decreases as the β_is increase, and almost disappears when LWE ciphertexts modulo $2N$ are used in all three blind rotations. Thus, we propose an ad-hoc method to remove the bias in the output noise by randomizing ACC between CMux operations. The randomization multiplies the i-th ACC by X^{ρ_i}. The exponents $\rho_0, \ldots, \rho_{n-2}$ are sampled independently and uniformly from $\mathbb{Z}_{2N}$, and ρ_{n-1} is determined by $\sum_{i\in[n]} \rho_i \equiv 0 \pmod{2N}$. The equation above ensures that the blind rotation result still encrypts the same plaintext. Experiments show that the noise is almost unbiased after applying this mitigation (see Figure 3e), and the mitigation overhead is negligible because it only requires a negacyclic rotation of the coefficients of ACC per external product.

[6] https://github.com/KAIST-CryptLab/refined-tfhe-lhe/tree/9b0426e

Table 13: Detailed parameters for EBS and Meta-PBS with $\log_2(t) \in [\![8, 12]\!]$. Blind rotation in EBS is performed in a ring dimension of kN, while Meta-PBS always uses $k = 1$. Within each parameter set below, all HomTruncRepeat* operations share the displayed B_{tr}. $c_{\text{meta}} = \log_2(q/(2t)) - \log_2(\text{erfc}^{-1}(p_{\text{fail}}) \cdot \sqrt{2V_{\text{in}}})$ measures the size of V_{in}.

t	Method	k	B_{br}	l_{br}	B_{ks}	l_{ks}	c_{Hom}	K	β	r	δ	T	ϵ	c_{meta}	B_{tr}	l_{tr}
2^8	EBS	8	2^{15}	2	2^5	5	3.92	/	/	/	/	/	/	/	/	/
	Meta-PBS	1	2^{15}	2	2^9	2	5.25	1	(9)	(16,144)	(66,71)	(74)	(78)	1.41	2^{23}	1
2^9	EBS	16	2^{15}	2	2^4	6	2.81	/	/	/	/	/	/	/	/	/
	Meta-PBS	1	2^{15}	2	2^7	3	4.24	2	(5,4)	(8,37,148)	(66,67,73)	(70,85)	(268,341)	1.33	2^{23}	1
2^{10}	EBS	32	2^{15}	2	2^3	8	1.46	/	/	/	/	/	/	/	/	/
	Meta-PBS	1	2^{15}	2	2^7	3	1.73	2	(6,6)	(4,23,138)	(66,66,68)	(68,77)	(204,186)	2.01	2^{23}	1
2^{11}	EBS	128	2^{15}	2	2^7	3	6.11	/	/	/	/	/	/	/	/	/
	Meta-PBS	1	2^{11}	3	2^4	6	6.38	2	(9,8)	(2,17,136)	(66,66,67)	(67,74)	(92,107)	2.40	2^{15}	2
2^{12}	EBS	256	2^{15}	2	2^7	3	3.77	/	/	/	/	/	/	/	/	/
	Meta-PBS	1	2^{15}	2	2^5	5	3.81	2	(14,12)	(1,14,168)	(66,66,83)	(66,73)	(13,23)	0.73	2^{15}	2

Table 14: Parameters for Int-LHE and Meta-WoP-PBS with $t = 2^8$. Some parameters exclusive to Int-LHE are not displayed, e.g., gadget parameters for HomTrace and RGSW ciphertexts.

Method	B_{br}	l_{br}	B_{ks}	l_{ks}	c_{Hom}	K	β, β_{CS}	r, r_{ext}	$\delta, \delta_{\text{CS}}$	T	$\epsilon, \epsilon_{\text{CS}}$	c_{meta}	B_{tr}	l_{tr}	ν	τ
LHE_4_4	2^{15}	2	2^4	3	NA	/	/	/	/	/	/	/	/	/	/	/
LHE_8	2^9	4	2^7	2	4.91	/	/	/	/	/	/	/	/	/	/	/
Meta_8C	2^{15}	2	2^2	10	6.68	2	(17,8) 25	(2,34,272) 4	(48,50,134) 48	(49,67)	(15,19) 22	2.13	2^{15}	2	2	3
Meta_8	2^{11}	3	2^2	10	9.89	2	(16,8) 10	(2,32,256) 10	(48,50,123) 48	(49,66)	(23,27) 57	2.18	2^{11}	3	2	3
Meta_8C-128	2^{11}	3	2^2	10	8.03	3	(8,7,3) 18	(2,16,112,336) 10	(85,85,97,163) 85	(86,93,153)	(77,57,39) 26	2.28	$(2^{12}, 2^{11}, 2^{11})$ 2^{11}	3	2	3
Meta_8-128	2^9	4	2^2	10	10.42	3	(8,7,3) 18	(2,16,112,336) 10	(85,85,97,163) 85	(86,93,153)	(77,57,39) 26	2.28	$(2^{12}, 2^{11}, 2^{11})$ 2^{11}	3	2	3

The lower-than-estimated output noise variance is not exclusive to Meta-PBS; it also appears in regular PBS. The measured standard deviation of one blind rotation is about 90.1% (or 78.6%) of the estimated one in tfhe-go (or tfhe-rs). Based on this measurement, the estimated ratio between the measured and expected noise standard deviation after three consecutive blind rotations would be $\sqrt{(1 + \frac{4}{5} + \frac{3}{4})/3} \cdot 90.1\% = 83.1\%$ (or 72.5%) in tfhe-go (or tfhe-rs).[7] This agrees with the measured ratios of 81.7% and 73.8%.

Understanding why blind rotations produce biased noise with a lower-than-expected standard deviation, and how implementations affect this behavior, remains an open problem.

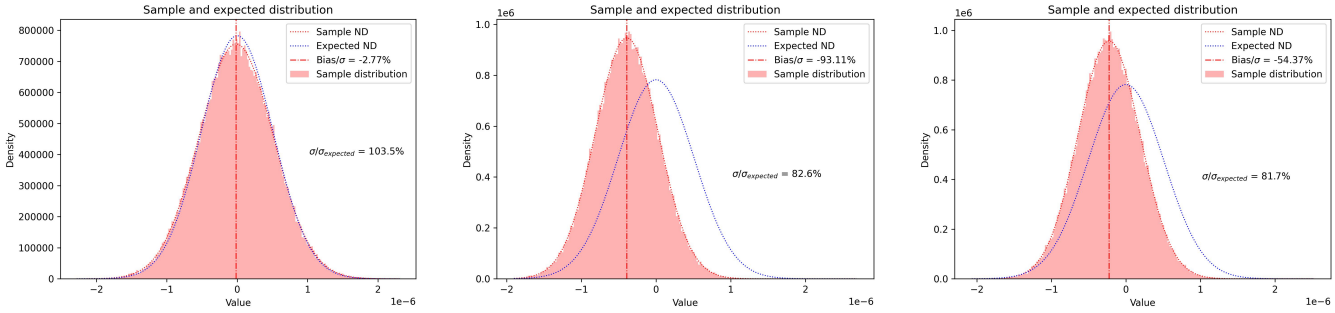
H.2 Independence of Output Noise Coefficients

Since TruncRepeat duplicates the coefficients of the noise polynomial in its input RLWE ciphertext, Meta-PBS should have correlated output noise coefficients. However, we argue that the output LWE ciphertexts of Meta-PBSManyLUTs still have independent noise terms.

This is because the duplicated noise coefficients correspond to the same LWE plaintext, and during sample extraction, at most one LWE ciphertext is extracted for the same LWE plaintext. Thus, the duplicated noise terms never appear in more than one output LWE ciphertext. If we additionally assume that blind rotation and TruncRepeat produce coefficient-wise independent noise coefficients, which is a common assumption for PBSManyLUTs-based algorithms like circuit bootstrapping, we can treat the output noise terms of Meta-PBSManyLUTs as independent.

As a final remark, although the duplication of the noise coefficients does not affect the independence analysis, it still introduces a slight drawback. Since the noise coefficients corresponding to the same LWE plaintext are highly correlated, the multi-value extraction method [13] cannot be applied to reduce the noise growth in post-bootstrap linear combinations.

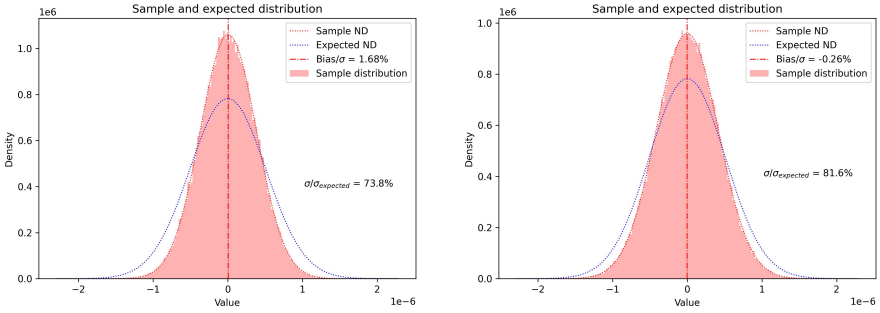
[7] The factors $\frac{4}{5}$ and $\frac{3}{4}$ are variance weights for the last two blind rotations: each coefficient of their input LWE ciphertexts is zero with probability $\frac{1}{5}$ and $\frac{1}{4}$, respectively.



(a) Meta-PBS in tfhe-go.

(b) Meta-PBS in tfhe-go with new CMux implementation.

(c) Three consecutive blind rotations in tfhe-go with new CMux implementation.



(d) Three consecutive blind rotations in tfhe-rs.

(e) Meta-PBS in tfhe-go with randomized blind rotation.

Figure 3: Noise distributions for Meta-PBS or three consecutive blind rotations in tfhe-go and tfhe-rs. The noise is normalized with respect to a ciphertext modulus of one. Each experiment is repeated 100,000 times.