Phase-Hiding Programmable Bootstrapping with Online Client Assistance

Xingke Chai

School of Mathematics and Statistics, Heilongjiang University, Harbin, China

Abstract

In Fast Fully Homomorphic Encryption over the Torus (TFHE), a client can go offline after uploading its ciphertexts and evaluation keys. The server uses the evaluation keys, which alone occupy 10.2 GiB under official 8-bit TFHE-rs Classic-Gaussian parameters ($N = 65536$), to evaluate lookup tables via blind rotation and retain refreshed ciphertexts for continued computation. When the key-holding client participates online, decrypting on the client to evaluate a server-private table in plaintext exposes both intermediate plaintexts and the table. We present Phase-Hiding Programmable Bootstrapping (PH-PBS), a client-assisted protocol suite that preserves the server's refreshed-ciphertext interface and replaces blind rotation, key switching, and their evaluation keys with two steady-state round trips per refresh while hiding intermediate plaintexts and server-private tables. In Round 1, the server applies centered modulus switching and the parties secret-share the discrete decryption phase outside any garbled circuit. Because rounding depends on the borrow between these shares, Round 2 runs a short garbled subtraction circuit logarithmic in table size to output a protected index while keeping private tables outside. For a single table (PH-PBS-Blind), the client re-encrypts only the selected blinded entry; for multiple same-input tables (PH-PBS-LWE and PH-PBS-GLWE), one Round-1 shift-encrypted token lets the server produce all output ciphertexts locally with online communication independent of table count. Across all 12 configurations (three modes at 2–8 bits), each party retains 1.08–3.28 MiB of persistent session state. At 8 bits, single-table PH-PBS-Blind takes 0.0595 ms of combined local computation and 3,726 B of online traffic, an 11.82× local speedup and a 94.6% traffic reduction over GC-PrivLUT, our same-interface garbled-lookup baseline. At 1 Gbps, our steady-state network model predicts lower end-to-end latency than single-threaded 8-bit sorted bootstrapping (SBS+CMS) for round-trip times below 194.97 ms. We prove full-session static semi-honest simulation security; under our grid-rounded Gaussian model and certified weight budgets over 1,024 refreshes and 64 terminal decodings, all 12 configurations achieve task failure bounds below 2^{-316} over random keys and execution for fixed non-adaptive tasks without mask-uniformity assumptions.

Keywords: Fully Homomorphic Encryption, TFHE, Programmable Bootstrapping, Client-Assisted Refresh, Private Lookup Tables, Garbled Circuits

1 Introduction

Since Gentry [Gen09] introduced fully homomorphic encryption (FHE), the mainstream outsourcing paradigm has let a client upload ciphertexts and evaluation keys and go offline while a server computes directly over encrypted data. Unlike packed schemes tailored to batched polynomial arithmetic, Fast Fully Homomorphic Encryption over the Torus (TFHE)

Email: xingkechai@gmail.com (Xingke Chai)

operates on individual learning-with-errors (LWE) ciphertexts and uses programmable bootstrapping (PBS) to evaluate an arbitrary discrete lookup table (LUT) while refreshing noise into a ciphertext ready for continued homomorphic computation [CGGI20, GBA21]. In mainstream non-interactive TFHE implementations such as official TFHE-rs [Zam26] and sorted bootstrapping with companion modulus switching (SBS+CMS) [BORLT26], the server independently runs key switching before modulus switching, blind rotation, and sample extraction (KS-then-PBS). Because the server lacks the client's secret decryption key, it must drive key switching and blind rotation using an uploaded key-switching key (KSK) and bootstrapping key (BSK). Both key size and refresh latency grow sharply with plaintext precision: official TFHE-rs 64-bit Classic-Gaussian evaluation keys (BSK + KSK) expand from roughly 121 MiB at 2–4 bits to 943.6 MiB at 6 bits, and reach 10.2 GiB at 8 bits ($N = 65536$) before key-generation working buffers (Section 6).

A natural first attempt is to accelerate non-interactive PBS on the server alone. Recent work advances this route along three directions: multi-bit bootstrapping [JP22] and SBS+CMS [BORLT26] prune external products during blind rotation, while mean compensation (DBM) [dRDV26] cancels approximation noise bias. A third line extends multi-function and larger-precision evaluation via multi-value bootstrapping [CIM19, GBA21], without-padding PBS and ManyLUT [CLOT21], and full-domain digit decomposition [MHW+24]. Despite these gains, non-interactive refresh still relies on BSK-driven blind rotation and hundred-MiB evaluation keys: even recent optimizations such as 4-bit DBM [dRDV26] and 4–8-bit SBS+CMS [BORLT26] retain 118.5–435.2 MiB of keys and require tens to hundreds of milliseconds per single-threaded refresh (up to 390.02 ms at 8 bits on our host). While high-end servers can absorb these overheads, storage, transfer, and compute costs remain heavy on resource-constrained lightweight servers like our 16 GB host.

Relieving the server of this burden therefore requires rethinking the offline-client assumption. In interactive outsourced services such as private inference over quantized models and proprietary rule scoring, the key-holding client already remains online waiting for query responses, so keeping all bootstrapping work on the server is no longer optimal. Enlisting the online client to assist refresh is a natural alternative. Building on FHE ciphertext sanitization [DS16] and functional re-encryption [AGHV25], Smart and Walter [SW25] construct an error-simulatable client-assisted TFHE re-encryption protocol. Under its decrypt-and-evaluate call pattern, however, two barriers persist. First, decrypting the ciphertext and evaluating the function in plaintext exposes both intermediate plaintexts and the evaluated function to the client, which rules out server-private LUTs. Second, although the server omits a BSK, error-simulatable sanitization still demands a massive exact KSK and heavy client-side key switching: in our same-host 4-bit reproduction, the client holds a 1,760.0 MiB exact KSK and spends 41.70 ms locally per non-terminal ciphertext-returning call (Section 6).

Once both intermediate plaintexts and server-private tables must stay hidden, client-assisted refresh naturally becomes a problem of secure two-party computation (2PC). In this adjacent area, hybrid homomorphic–2PC inference frameworks such as GAZELLE [JVC18], Delphi [MLS+20], and Cheetah [HjLHD22] target mostly public fixed activations and switch representations between homomorphic linear layers and secret-shared nonlinear layers. Dedicated 2PC LUT protocols, including FLUTE [BHS+23] and ROTL [HLL+26] for public tables and FABLE [SPBZ25] for private tables, likewise accept and return secret shares. Real outsourced inference and rule-scoring workloads, however, rarely stop at an isolated lookup: they execute a multi-input, multi-layer computation graph, modeled as a directed acyclic graph (DAG), that alternates server-side local affine combinations with private LUT refreshes, where one ciphertext's output can fan out into multiple parallel private tables and later reconverge in downstream linear layers. When the server retains native LWE ciphertexts to run these multi-input affine layers locally, integrating

share-based 2PC lookup protocols requires extra conversions between secret shares and ciphertexts. Conversely, keeping both the native LWE ciphertext interface and dual privacy via a direct Boolean garbled multiplexer tree scales gates and communication sharply with table size, bit-width, and parallel table count: our self-built same-interface baseline GC-PrivLUT requires 2,065 non-free AND gates and 68,976 B of online communication for a single 8-bit table refresh (Section 6).

Direct garbled lookup is expensive because it places private table selection inside a garbled circuit, and an end-to-end garbled formulation would additionally pull the high-dimensional key inner product inside. In fact, the operation splits into three parts with very different costs: linear decryption-phase computation (a high-dimensional secret-key inner product compressed by modulus switching), nonlinear phase rounding to isolate the plaintext index, and private table selection that outputs a refreshed ciphertext. Only the tiny nonlinear rounding step genuinely requires a Boolean garbled circuit; placing either the high-dimensional key inner product or the private table inside the circuit is unnecessary. Guided by this separation, we present PH-PBS, a phase-hiding programmable bootstrapping suite for interactive multi-input affine–LUT DAG workloads with an online key-holding client. PH-PBS retains server-side modulus switching and native LWE ciphertexts, removes BSK-driven blind rotation, KSK key switching, and their evaluation keys at a steady-state cost of two network round trips (2-RTT) per refresh, and hides both intermediate plaintexts and server-private tables. In Round 1, server-side centered modulus switching (CtrMS) compresses the decryption phase onto a small discrete grid, and the two parties additively secret-share this discrete phase outside any garbled circuit via secret-bound pseudorandom pads (PhasePads); this completes the high-dimensional key inner product with zero online oblivious transfers and zero garbled gates while revealing neither the phase nor the plaintext. Because subtracting the two shares creates a low-bit borrow into the index bits, neither party can round its share locally. Round 2 therefore runs only a short garbled subtraction circuit whose size scales logarithmically with table size and stays independent of ciphertext dimension and table count; this circuit outputs a lookup index protected by wire labels or a random shift while leaving every private table outside.

From this protected index outside the circuit, PH-PBS completes table evaluation and ciphertext refresh through two workload-specialized backends. For a single table, PH-PBS-Blind lets the client use the active wire labels to decrypt and freshly re-encrypt only the selected blinded entry, which the server unblinds into a refreshed native LWE ciphertext. When a DAG node fans out one ciphertext into multiple parallel private tables on the same input, PH-PBS-LWE and PH-PBS-GLWE instead let the server combine the shift-masked index with a single-use Round-1 shift-encrypted token to evaluate all tables locally with online communication independent of the table count. Between these two token modes, PH-PBS-LWE operates in small-dimension LWE space without domain restrictions, whereas PH-PBS-GLWE packs the token into a single rank-1 generalized LWE (GLWE) monomial ciphertext for smaller fresh noise under a half-domain input contract. When the DAG reaches its designated final outputs, all three modes omit the fourth message via a 1.5-RTT terminal decoding interface that delivers only the rounded output plaintexts without exposing unrounded noise phases.

1.1 Contributions

- **Blind-rotation-free 2-RTT protocol suite and exact projected GLWE evaluation (Section 3).** We construct the two-round PH-PBS suite (PH-PBS-Blind, PH-PBS-LWE, and PH-PBS-GLWE) together with 1.5-RTT terminal decoding to preserve native server-side LWE ciphertexts for multi-input affine–LUT DAG execution while eliminating BSK-driven blind rotation and KSK key switching. For PH-PBS-GLWE, we prove exact coefficientwise equivalence between projected and full-polynomial evaluation under identical masks and active body coordinates, and

prove that on-demand Gaussian noise sampling on active coordinates is identically distributed to full-polynomial restriction, accelerating client token generation by 2.10–4.79× across 2–8 bits.

- **Discrete correctness certificates with shared-token reconvergence and full-session simulation security (Section 4–Section 5).** We establish an exact discrete CtrMS integer identity and a sufficient correctness condition when outputs sharing a multi-table token reconverge in downstream DAG affine layers. Under our grid-rounded Gaussian model and certified weight budgets over 1,024 refreshes and 64 terminal decodings, all 12 configurations (three modes at 2–8 bits) achieve task failure bounds below 2^{-698} under ideal mask residues, below 2^{-316} over random keys and execution for fixed non-adaptive tasks without mask-uniformity assumptions, and below 2^{-128} per fixed key under worst-case masks (with conservative budgets for PH-PBS-GLWE). We prove full-session simulation security for both views under static semi-honest corruptions and public execution schedules, supported by minimum classical lattice attack cost estimates of 132.00 bits (MATZOV) and 134.42 bits (BDGL16) across all five native-grid lattice instances.
- **Rust implementation and same-host multi-axis evaluation (Section 6).** We implement all 12 configurations in Rust, holding each party's persistent session state to 1.08–3.28 MiB (0.12 s one-time setup). At 8-bit precision, single-table PH-PBS-Blind requires 0.0595 ms of combined local compute and 3,726 B of online traffic (11.82× faster locally and 94.6% less communication than same-interface GC-PrivLUT), and achieves a 194.97 ms 1 Gbps steady-state break-even RTT against single-threaded 8-bit SBS+CMS (7.78–17.41 ms at 4 bits). Across 1 to 16 parallel same-input private tables, PH-PBS-LWE and PH-PBS-GLWE hold online communication strictly constant (4,460 B and 7,051 B at 8 bits); across 8 concurrent session pairs, PH-PBS-Blind sustains 34.2k–38.5k ops/s, and all three modes pass end-to-end validation on 20-layer multi-input affine–LUT DAGs and 1,400-layer sustained COT refill runs.

2 Preliminaries and Problem Formulation

2.1 Notation, Encoding, and Ciphertext Representations

Table 1 summarizes the notation used throughout the paper. Let $q = 2^{64}$ denote the native ciphertext modulus and write $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. For any power-of-two modulus R and any integer or residue class x, we write $[x]_R$ for its unique representative in $\{0, \dots, R-1\}$ and $\mathrm{ctr}_R(x)$ for its centered integer representative in $[-R/2, R/2)$. Polynomial arithmetic takes place in the negacyclic ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N+1)$ with $N = 2048$; because $X^N = -1$ and $X^{2N} = 1$, multiplication by a monomial acts as a negacyclic shift that flips coefficient signs across multiples of N (for example, $X^{-1} = -X^{N-1}$). For plaintext precision $k \in \{2, 4, 6, 8\}$, let $p = 2^k$ and define the message space $\mathcal{M} = \{0, \dots, p-1\}$. We use one-padding-bit integer encoding with scaling factor $\Delta = q/(2p) = 2^{63-k}$, which maps $m \in \mathcal{M}$ to $[\Delta m]_q \in [0, q/2)$.

Within the TFHE instantiation [CGGI20] of learning with errors (LWE) [Reg09], our protocol uses an independently and uniformly sampled binary secret $\mathbf{s} \in \{0,1\}^d$ together with a zero-mean integer error distribution χ_σ induced by a finite-precision Gaussian sampler of normalized standard deviation σ rounded to the mode's native ciphertext grid $\mathbb{Z}_{q_{\mathrm{grid}}}$ (modeled explicitly in Section 4). In the ideal mask model, to encrypt $m \in \mathcal{M}$, the client samples the mask uniformly on $\mathbb{Z}^d_{q_{\mathrm{grid}}}$, independently of $\mathbf{s}$ and the native-grid encryption error $e_{\mathrm{grid}} \leftarrow \chi_\sigma$, and represents the resulting ciphertext $\mathbf{c} = (\mathbf{a}, b) \in \mathbb{Z}_q^d \times \mathbb{Z}_q$ over $\mathbb{Z}_q$ via the exact lift $x \mapsto (q/q_{\mathrm{grid}})x \bmod q$, with body $b = [\langle \mathbf{a}, \mathbf{s}\rangle + \Delta m + e]_q$ where $e = (q/q_{\mathrm{grid}})e_{\mathrm{grid}}$ denotes the lifted error. Its decryption phase is $\mathrm{Phase}_{\mathbf{s}}(\mathbf{c}) = [b - \langle \mathbf{a}, \mathbf{s}\rangle]_q =$

Table 1: Core notation used throughout the paper.

Symbol	Meaning
λ, q	Security parameter ($\lambda = 128$) and native ciphertext modulus ($q = 2^{64}$)
$[x]_R, \mathrm{ctr}_R(x)$	Representatives of x modulo R in $\{0, \dots, R-1\}$ and $[-R/2, R/2)$
$k, p, \mathcal{M}, \Delta$	Plaintext bit-width $k \in \{2, 4, 6, 8\}$, size $p = 2^k$, space $\mathcal{M} = \{0, \dots, p-1\}$, scale $\Delta = q/(2p)$
$d, n, N, \mathcal{R}_q$	Active LWE dimension d (small dimension n or degree $N = 2048$) and ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$
$\mathbf{s}, s(X), \mathbf{s}_{\mathrm{ext}}$	Binary LWE secret $\mathbf{s} \in \{0,1\}^d$, binary polynomial secret $s(X) \in \mathcal{R}_q$, and coefficient vector $\mathbf{s}_{\mathrm{ext}} \in \{0,1\}^N$
$\mathbf{c}, C, \chi_\sigma$	LWE ciphertext $\mathbf{c} = (\mathbf{a}, b)$, rank-1 GLWE ciphertext $C = (A, B)$, and grid-rounded Gaussian integer error distribution χ_σ of normalized scale σ
$R_{\mathrm{sh}}, \tau_{\mathrm{sh}}, b_{\mathrm{ms}}, b_{\mathrm{sh}}$	Centered modulus switching (CtrMS) modulus $R_{\mathrm{sh}} = 2^{b_{\mathrm{ms}}} = 2p \cdot \tau_{\mathrm{sh}}$, resolution τ_{sh}, and phase-sharing / subtraction bit-width b_{sh}
$\tau_{\mathrm{rot}}, b_B, \Delta_{\mathrm{GC}}$	GLWE lookup stride $\tau_{\mathrm{rot}} = N/p$, token body precision b_B, and global garbled-circuit / correlated-OT offset $\Delta_{\mathrm{GC}} \in \{0,1\}^\lambda$
r, α, M_v	Client one-time random shift r, server-visible shift-masked discrete index α, and parallel private LUT count M_v at node v
$\mathcal{C}, \mathcal{S}, \mathcal{G}, W_1, W_2$	Client $\mathcal{C}$, server $\mathcal{S}$, public execution schedule $\mathcal{G}$, and certified multi-input affine weight budgets (W_1, W_2)

$[\Delta m + e]_q$, and standard round-half-up decoding computes $\mathrm{Dec}_{\mathbf{s}}(\mathbf{c}) = [\lfloor \mathrm{Phase}_{\mathbf{s}}(\mathbf{c})/\Delta + 1/2 \rfloor]_p$, which recovers m whenever $|\mathrm{ctr}_q(e)| < \Delta/2$. Relative to a prescribed message m, we define the normalized phase error $e_{\mathrm{norm}}(\mathbf{c}, m) = q^{-1}\mathrm{ctr}_q(\mathrm{Phase}_{\mathbf{s}}(\mathbf{c}) - [\Delta m]_q)$ to measure the centered deviation of the unrounded phase. Given LWE ciphertexts $\{\mathbf{c}_h\}$ under the same secret $\mathbf{s}$, integer weights $\{w_h\}$, and an integer bias β, the homomorphic affine combination $\mathbf{c}_{\mathrm{aff}} = \sum_h w_h \mathbf{c}_h + (\mathbf{0}^d, [\Delta\beta]_q) \pmod q$ satisfies $\mathrm{Phase}_{\mathbf{s}}(\mathbf{c}_{\mathrm{aff}}) = [\Delta(\beta + \sum_h w_h m_h) + \sum_h w_h e_h]_q$, which preserves the linear message relation while accumulating weighted errors.

For ring learning with errors (rank-1 GLWE) [LPR10, CGGI20], the client samples a binary-coefficient secret polynomial $s(X) = \sum_{i=0}^{N-1} s_i X^i \in \mathcal{R}_q$ with coefficient vector $\mathbf{s}_{\mathrm{ext}} = (s_0, \dots, s_{N-1}) \in \{0,1\}^N$. A GLWE encryption of a message polynomial $P(X) \in \mathcal{R}_q$ is a pair $C = (A, B) \in \mathcal{R}_q^2$, where $A \in \mathcal{R}_q$ is uniform and each coefficient of $E \in \mathcal{R}_q$ is drawn from χ_σ, all mutually independently and independently of $s(X)$, with $B = As + P + E \in \mathcal{R}_q$; its phase polynomial is $\mathrm{Phase}_s(C) = B - As = P + E$.

2.2 Modulus Switching, Sample Extraction, and Polynomial Lookup

Coefficientwise modulus switching compresses a modulus-q LWE ciphertext to a smaller power-of-two modulus $R_{\mathrm{sh}} \mid q$ [CGGI20]. For $x \in \mathbb{Z}_q$, define the round-half-up map $\mathrm{MS}_{q \to R_{\mathrm{sh}}}(x) = [\lfloor (R_{\mathrm{sh}}/q)[x]_q + 1/2 \rfloor]_{R_{\mathrm{sh}}}$; applying it componentwise to $\mathbf{c} = (\mathbf{a}, b)$ yields $\hat{\mathbf{c}} = (\hat{\mathbf{a}}, \hat{b})$ with switched discrete phase $[\hat{b} - \langle \hat{\mathbf{a}}, \mathbf{s} \rangle]_{R_{\mathrm{sh}}}$. Following the mean-compensation principle [dRDV26], Section 3 augments this body conversion with discrete centered modulus switching (CtrMS) to cancel the binary-key mean bias, and Section 4 establishes its exact integer identity and error bounds.

Sample extraction maps the constant phase coefficient of a GLWE ciphertext into an N-dimensional LWE ciphertext without adding approximation noise [CGGI20]. For $C = (A, B) \in \mathcal{R}_q^2$ with $A(X) = \sum_{i=0}^{N-1} a_i X^i$ and $B(X) = \sum_{i=0}^{N-1} b_i X^i$, define $\mathrm{Ext}_0(C) = ((a_0, -a_{N-1}, \dots, -a_1), b_0) \in \mathbb{Z}_q^N \times \mathbb{Z}_q$, with negations taken modulo q. By negacyclic

multiplication, this extraction satisfies $\mathrm{Phase}_{\mathbf{s}_{\mathrm{ext}}}(\mathrm{Ext}_0(C)) = [[X^0](B - As)]_q$, where $[X^0](\cdot)$ denotes the constant-term coefficient.

Polynomial lookup pairs an encrypted monomial with sample extraction to read a private table entry [GBA21]. Assume $p \mid N$ and set the stride $\tau_{\mathrm{rot}} = N/p$. Given any function $f : \mathcal{M} \to \mathcal{M}$, encode its entries into $Q_f(X) = \sum_{h=0}^{p-1} f(h) X^{\tau_{\mathrm{rot}} h} \in \mathcal{R}_q$. For every $m \in \mathcal{M}$, only the term $h = m$ in $Q_f(X) X^{-\tau_{\mathrm{rot}} m}$ has exponent zero modulo N, so $[X^0](Q_f(X) X^{-\tau_{\mathrm{rot}} m}) = f(m)$. Hence, if a GLWE ciphertext C has phase $\Delta X^{-\tau_{\mathrm{rot}} m} + E$, then $\mathrm{Ext}_0(Q_f C)$ satisfies $\mathrm{Phase}_{\mathbf{s}_{\mathrm{ext}}}(\mathrm{Ext}_0(Q_f C)) = [\Delta f(m) + [X^0](Q_f E)]_q$ and places the table output directly into a native LWE ciphertext.

2.3 Two-Party Secure Computation Primitives

To secret-share the decryption phase without online OTs and securely round the two shares in Round 2, PH-PBS invokes two standard semi-honest two-party primitives: oblivious transfer (OT) and Boolean garbled circuits (GC). In 1-out-of-2 OT, a sender holds (M_0, M_1) and a receiver holds a choice bit $c \in \{0,1\}$; the protocol delivers M_c to the receiver without revealing c to the sender or M_{1-c} to the receiver. During session setup, PH-PBS uses OT extension [IKNP03, ALSZ17] (seeded by $\lambda = 128$ base OTs) to distribute PhasePads PRF keys, and separately instantiates a random-choice correlated OT (COT) extension to supply Round-2 garbled-circuit input labels in bulk. In this COT interface, the sender holds a private global offset $\Delta_{\mathrm{GC}} \in \{0,1\}^\lambda$ with $\mathrm{lsb}(\Delta_{\mathrm{GC}}) = 1$; each COT instance outputs a uniform block $Q_i \in \{0,1\}^\lambda$ to the sender and a uniform bit $c_i^* \in \{0,1\}$ with correlated block $T_i = Q_i \oplus (c_i^* \Delta_{\mathrm{GC}})$ to the receiver. In PH-PBS, the server acts as the COT sender and the client acts as the receiver.

For Boolean circuits with public topology, we instantiate Free-XOR [KS08] and Half-Gates [ZRE15] garbled circuits with a tweakable circular correlation-robust hash function [GKW+20]. Each wire w carries a pair of λ-bit labels (L_w^0, L_w^1) satisfying $L_w^1 = L_w^0 \oplus \Delta_{\mathrm{GC}}$; since $\mathrm{lsb}(\Delta_{\mathrm{GC}}) = 1$, the two labels have opposite least significant bits (selection bits). For an output wire w with decoding bit $d_w = \mathrm{lsb}(L_w^0)$, an active output label ℓ_w determines the plaintext bit $y_w = \mathrm{lsb}(\ell_w) \oplus d_w$; Section 3 specifies how each PH-PBS backend and terminal decoding consume these output wires.

2.4 Multi-Input DAG Task Model and Security Goals

We consider interactive outsourced computation between a key-holding client $\mathcal{C}$ and an evaluation server $\mathcal{S}$. The client $\mathcal{C}$ privately holds the mode-appropriate secret key and inputs $\mathbf{x} = (x_1, \ldots, x_{N_{\mathrm{in}}}) \in \mathcal{M}^{N_{\mathrm{in}}}$, whose initial encryptions are registered on the server under output handles $(0,1), \ldots, (0, N_{\mathrm{in}})$. Specifically, PH-PBS-Blind and PH-PBS-LWE operate under the small binary LWE secret $\mathbf{s} \in \{0,1\}^n$, whereas PH-PBS-GLWE uses $s(X) \in \mathcal{R}_q$ for token generation while its initial ciphertexts, sample-extracted refresh outputs, and downstream affine combinations all remain native N-dimensional LWE ciphertexts under $\mathbf{s}_{\mathrm{ext}} \in \{0,1\}^N$. The server $\mathcal{S}$ privately holds integer affine weights and biases $\{(w_{v,h}, \beta_v)\}$ together with lookup tables $\{f_{v,t} : \mathcal{M} \to \mathcal{M}\}$. A public execution schedule $\mathcal{G}$ specifies the directed acyclic graph (DAG) topology, bit-width k, evaluation mode (PH-PBS-Blind, PH-PBS-LWE, or PH-PBS-GLWE), total refresh count L, terminal output count N_{out}, the predecessor handle set $\mathrm{Pred}(v)$ and parallel private table count $M_v \geq 1$ at each refresh node $v \in \{1, \ldots, L\}$, and certified affine weight budgets (W_1, W_2). Because an upstream refresh node v' may emit $M_{v'}$ ciphertexts in parallel, each predecessor handle $h = (v', t) \in \mathrm{Pred}(v)$ identifies a specific output ciphertext $\mathbf{c}_{v',t}$.

At node v, the server locally computes the multi-input homomorphic affine combination $\mathbf{c}_v^{\mathrm{in}} = \sum_{h \in \mathrm{Pred}(v)} w_{v,h} \mathbf{c}_h + (\mathbf{0}^d, [\Delta \beta_v]_q) \pmod q$. Because parallel outputs evaluated from one multi-table token share correlated token noise, we partition $\mathrm{Pred}(v)$ by encryption-noise

source into groups $\mathcal{P}_v$: each independently encrypted initial input and each freshly re-encrypted PH-PBS-Blind output forms a singleton group, whereas outputs evaluated from the same PH-PBS-LWE or PH-PBS-GLWE token form one group. For each $G \in \mathcal{P}_v$, define $\overline{w}_{v,G} = \sum_{h \in G} |w_{v,h}|$; the local affine weights must satisfy the ℓ_1 budget $\sum_{G \in \mathcal{P}_v} \overline{w}_{v,G} \le W_1$ (bounding deterministic rounding and quantization shifts) and the source-coalesced ℓ_2 budget $\sum_{G \in \mathcal{P}_v} \overline{w}_{v,G}^2 \le W_2^2$ (bounding aggregated Gaussian noise variance; Section 4). The two parties then evaluate the M_v server-private LUTs on the plaintext μ_v encoded in $\mathbf{c}_v^{\text{in}}$ and register the resulting M_v native LWE ciphertexts under new handles $(v,1),\dots,(v,M_v)$ that $\mathcal{S}$ retains for downstream computation. Under the input-domain contract, PH-PBS-Blind and PH-PBS-LWE reduce affine inputs automatically modulo p and support arbitrary functions $\mathbb{Z}_p \to \mathbb{Z}_p$, whereas PH-PBS-GLWE preserves the negacyclic structure of $\mathcal{R}_q$ ($X^N = -1$) and therefore follows the standard TFHE one-padding-bit half-domain contract $\mu_v \in \{0,\dots,p-1\}$.

Our security target is full-session simulation security against a static semi-honest adversary [Gol04] that corrupts either $\mathcal{C}$ or $\mathcal{S}$ prior to execution and follows the protocol honestly. Given the public execution schedule $\mathcal{G}$, a corrupted client learns nothing beyond its own inputs and the final terminal plaintexts (hiding all intermediate plaintexts, server affine parameters, and private LUTs), and a corrupted server learns nothing beyond its own affine parameters and private LUTs (hiding the client's secret key, inputs, intermediate plaintexts, and terminal outputs). Sessions enforce strict single-use state discipline: neither one-time ciphertext tokens nor consumed COT tuples may ever be reused across invocations. We formalize the full-session ideal functionality $\mathcal{F}_{\text{PH-PBS}}$ (covering setup, multi-input affine evaluation, single- and multi-table refresh, and terminal decoding), exact allowed leakage, and dual-view simulation proofs in Section 5.

3 The PH-PBS Protocol Suite

3.1 Protocol Overview and Session Setup

Figure 1 depicts the end-to-end execution flow across all three PH-PBS modes. At any DAG refresh node v (Subsection 2.4), each refresh runs in two steady-state network round trips: **Round 1** (Stages 1–2) applies CtrMS on a b_{ms}-bit grid and uses PhasePads to secret-share the resulting discrete phase $j = [u-\eta]_{2^{b_{\text{sh}}}}$ at the mode-specific width b_{sh} outside any circuit, while **Round 2** (Stages 3–4) runs a short $(b_{\text{sh}}-1)$-AND garbled subtraction circuit solely to resolve the rounding borrow and emit a protected index (Subsection 3.2). Outside the circuit, PH-PBS-Blind (Subsection 3.3) binds blinded single-table lookup to the active output wire labels with fresh seeded LWE re-encryption, whereas PH-PBS-LWE and projected PH-PBS-GLWE (Subsection 3.4) coalesce one single-use shift token into Stage 2 so $\mathcal{S}$ evaluates all M_v parallel tables locally at constant online communication; for designated final DAG outputs, 1.5-RTT terminal decoding (Subsection 3.5) omits Stage 4 and delivers only the rounded plaintext to $\mathcal{C}$.

During one-time session setup, $\mathcal{C}$ samples the mode secret key of Subsection 2.4 ($\mathbf{s} \in \{0,1\}^n$, or $s(X) \in \mathcal{R}_q$ with coefficient vector $\mathbf{s}_{\text{ext}} \in \{0,1\}^N$) and uploads the initial encrypted inputs under handles $(0,1),\dots,(0,N_{\text{in}})$. The two parties then run OT extension to deliver one of two PhasePads PRF keys per secret-key bit, initialize the garbling and COT state (carrying $\mathcal{S}$'s global offset Δ_{GC} and a shared PRG seed for zero-bandwidth server circuit inputs), and prefill the initial COT pool of capacity C_{pool}.

Online execution follows the public DAG schedule $\mathcal{G}$: after each local multi-input affine combination at node v, one refresh advances the PhasePads counter, consumes b_{sh} COT instances (plus one unused token in token modes), and registers M_v refreshed native LWE handles. Whenever the COT pool holds at least b_{sh} instances and a prepared token is ready, each refresh completes in strictly 2 RTT; if the COT pool is depleted before an

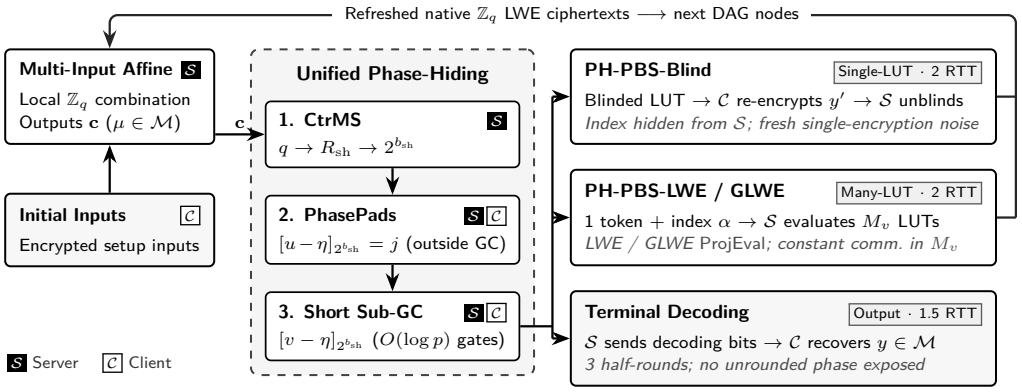

Figure 1: End-to-end PH-PBS architecture. Multi-input DAG affine evaluation, unified two-round phase-hiding frontend, outside-circuit single- and multi-table backends, and 1.5-RTT terminal decoding.

invocation, the parties run an inline OT-extension refill before Stage 1. Implementation details for mask kernels, PRG expansion, and streaming refill queues appear in Section 6.

3.2 Unified Phase-Hiding Stage

The unified phase-hiding frontend maps a modulus-q LWE ciphertext into a protected discrete lookup index across Stages 1–3 in three steps: centered modulus switching, off-circuit additive phase sharing via PhasePads, and short garbled subtraction.

Centered modulus switching (CtrMS). Given a native modulus-q LWE ciphertext $\mathbf{c} = (\mathbf{a}, b) \in \mathbb{Z}_q^d \times \mathbb{Z}_q$ encrypting $\mu \in \mathcal{M}$, $\mathcal{S}$ first applies CtrMS to target modulus $R_{\mathrm{sh}} = 2^{b_{\mathrm{ms}}} = 2p \cdot \tau_{\mathrm{sh}}$ (so $b_{\mathrm{ms}} = k + 1 + \log_2 \tau_{\mathrm{sh}}$) with step size $h_{\mathrm{ms}} = q/R_{\mathrm{sh}} = 2^{64-b_{\mathrm{ms}}}$. For each mask coordinate $i \in \{0, \ldots, d-1\}$, $\mathcal{S}$ computes $\hat{a}_i = \mathrm{MS}_{q \to R_{\mathrm{sh}}}(a_i)$, its signed quantization residual $\varepsilon_i = \mathrm{ctr}_q(h_{\mathrm{ms}}\hat{a}_i - a_i)$, and $\nu_i = \mathrm{trunc}(\varepsilon_i/2)$. Because binary secrets have mean $\mathbb{E}[s_i] = 1/2$, uncompensated rounding biases the switched phase by $-\frac{1}{2}\sum_{i=0}^{d-1} \varepsilon_i$. Following the mean-compensation principle [dRDV26] to cancel this bias and incorporate a half-step centering pre-shift, $\mathcal{S}$ computes the compensated body $\hat{b} = \mathrm{MS}_{q \to R_{\mathrm{sh}}}([b + \delta_{\mathrm{ms}}(\mathbf{a})]_q)$ using the integer offset

$$\delta_{\mathrm{ms}}(\mathbf{a}) = \sum_{i=0}^{d-1} \nu_i - \mathrm{trunc}\left(\frac{1}{2}\sum_{i=0}^{d-1}(2\nu_i - \varepsilon_i)\right) - \frac{h_{\mathrm{ms}}}{2}. \tag{1}$$

Since $2\sum_{i=0}^{d-1} \nu_i - \sum_{i=0}^{d-1}(2\nu_i - \varepsilon_i) = \sum_{i=0}^{d-1} \varepsilon_i$, the offset $\delta_{\mathrm{ms}}(\mathbf{a})$ equals $\frac{1}{2}\sum_{i=0}^{d-1} \varepsilon_i - h_{\mathrm{ms}}/2$ exactly when $\sum_{i=0}^{d-1} \varepsilon_i$ is even and differs from it by at most $1/2$ when odd; Section 4 establishes the exact CtrMS integer phase identity and error bounds. Next, $\mathcal{S}$ truncates $(\hat{\mathbf{a}}, \hat{b})$ to the mode-specific share width b_{sh}. In PH-PBS-Blind and PH-PBS-LWE, whenever the discrete phase error satisfies $-\tau_{\mathrm{sh}}/2 \le \tilde{e} < \tau_{\mathrm{sh}}/2$, truncating to $b_{\mathrm{sh}} = b_{\mathrm{ms}} - 1$ (so $2^{b_{\mathrm{sh}}} = p\tau_{\mathrm{sh}}$) drops the redundant most significant bit without wraparound ambiguity and saves one bit of sharing and circuit width. In PH-PBS-GLWE, $\mathcal{S}$ retains the full width $b_{\mathrm{sh}} = b_{\mathrm{ms}}$ (so $2^{b_{\mathrm{sh}}} = 2p\tau_{\mathrm{sh}}$) to preserve the negacyclic sign bit modulo $2p$ over $\mathcal{R}_q$. We write $j = [\hat{b} - \langle \hat{\mathbf{a}}, \mathbf{s} \rangle]_{2^{b_{\mathrm{sh}}}}$ for the resulting discrete phase (using $\mathbf{s}_{\mathrm{ext}}$ in GLWE mode).

Off-circuit additive phase sharing (PhasePads, Stage 1). To compute the d-dimensional inner product $\langle \hat{\mathbf{a}}, \mathbf{s} \rangle$ outside any garbled circuit, the parties initialize PhasePads once during

setup: they run one 1-out-of-2 OT per secret bit $s_i \in \{0,1\}$ (instantiated via OT extension) so that $\mathcal{C}$ receives only the pseudorandom function (PRF) key K_{i,s_i}, while $\mathcal{S}$ retains both $(K_{i,0}, K_{i,1})$. On each online invocation with fresh counter ctr (bound to the session identifier sid), both parties derive per-coordinate PRF pads $P_{i,\iota} = [\mathrm{PRF}_{K_{i,\iota}}(\mathrm{sid} \,\|\, \mathrm{ctr})]_{2^{b_{\mathrm{sh}}}}$ for each held key ($\iota \in \{0,1\}$). The server samples a uniform share $\eta \in \mathbb{Z}_{2^{b_{\mathrm{sh}}}}$ and transmits the scalar offset γ and correction vector $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_{d-1})$ defined by

$$\gamma = \left[\hat{b} + \eta - \sum_{i=0}^{d-1} P_{i,0}\right]_{2^{b_{\mathrm{sh}}}}, \qquad \xi_i = \left[P_{i,0} - P_{i,1} - \hat{a}_i\right]_{2^{b_{\mathrm{sh}}}}. \tag{2}$$

Using only its keys $\{K_{i,s_i}\}_{i=0}^{d-1}$, $\mathcal{C}$ evaluates $u = [\gamma + \sum_{i=0}^{d-1}(P_{i,s_i} + s_i\xi_i)]_{2^{b_{\mathrm{sh}}}}$. Because $P_{i,s_i} + s_i\xi_i \equiv P_{i,0} - s_i\hat{a}_i \pmod{2^{b_{\mathrm{sh}}}}$ holds identically for both $s_i = 0$ and $s_i = 1$, the pads $P_{i,0}$ cancel and $\mathcal{C}$ obtains $u \in \mathbb{Z}_{2^{b_{\mathrm{sh}}}}$ satisfying

$$[u - \eta]_{2^{b_{\mathrm{sh}}}} = j. \tag{3}$$

This computes the entire d-dimensional secret inner product with zero online OTs and zero garbled gates. Under the PRF assumption, every unselected pad $P_{i,1-s_i}$ masks ξ_i and the uniform share η masks γ, so the Stage 1 message and phase-share view are computationally indistinguishable from uniform and reveal neither j nor its unrounded error (Section 5).

Short garbled subtraction and index rounding (Stages 2–3). Under the correctness condition of Section 4, the discrete phase takes the form $j = [\tau_{\mathrm{sh}}\mu + \tilde{e}]_{2^{b_{\mathrm{sh}}}}$ with $-\tau_{\mathrm{sh}}/2 \le \tilde{e} < \tau_{\mathrm{sh}}/2$; adding the half-step $\tau_{\mathrm{sh}}/2$ and dropping the low $\log_2 \tau_{\mathrm{sh}}$ bits therefore rounds off $\tilde{e}$ and isolates the target index. Because subtracting η from $u + \tau_{\mathrm{sh}}/2$ propagates a low-bit borrow into the upper index bits, neither party can round locally. Instead, $\mathcal{C}$ locally folds $\tau_{\mathrm{sh}}/2$ and any mode-specific random shift into one b_{sh}-bit circuit input $v \in \mathbb{Z}_{2^{b_{\mathrm{sh}}}}$: PH-PBS-Blind and terminal decoding set $v = [u + \tau_{\mathrm{sh}}/2]_{2^{b_{\mathrm{sh}}}}$; PH-PBS-LWE (with token shift $r \in \mathbb{Z}_p$) sets $v = [u + \tau_{\mathrm{sh}}/2 - \tau_{\mathrm{sh}} r]_{2^{b_{\mathrm{sh}}}}$; and PH-PBS-GLWE decomposes its rotation shift $r \in \mathbb{Z}_{2N}$ as $r = \tau_{\mathrm{rot}} q_r + r_{\mathrm{lo}}$ with $q_r = \lfloor r/\tau_{\mathrm{rot}} \rfloor \in \mathbb{Z}_{2p}$ and $r_{\mathrm{lo}} = [r]_{\tau_{\mathrm{rot}}} \in \mathbb{Z}_{\tau_{\mathrm{rot}}}$, then sets $v = [u + \tau_{\mathrm{sh}}/2 - \tau_{\mathrm{sh}} q_r]_{2^{b_{\mathrm{sh}}}}$. In Stage 2, $\mathcal{C}$ consumes b_{sh} precomputed COT tuples (c_w^*, T_w) (Subsection 2.3), retains each T_w as its active input wire label for bit v_w, and sends the choice-correction bits $\mathsf{corr}_w = v_w \oplus c_w^*$ ($w \in \{0, \ldots, b_{\mathrm{sh}} - 1\}$); $\mathcal{S}$ sets the corresponding client input zero-labels to $L_w^0 = Q_w \oplus (\mathsf{corr}_w \Delta_{\mathrm{GC}})$, which satisfies $T_w = L_w^0 \oplus (v_w \Delta_{\mathrm{GC}})$. For its own private input η, $\mathcal{S}$ avoids online input-label transfer by deriving b_{sh} blocks $\{\ell_w^{(\eta)}\}$ synchronously with $\mathcal{C}$ from the setup shared PRG seed as $\mathcal{C}$'s active input labels and defining its input zero-labels locally as $L_w^{(\eta),0} = \ell_w^{(\eta)} \oplus (\eta_w \Delta_{\mathrm{GC}})$. In Stage 3, $\mathcal{S}$ transmits Free-XOR [KS08] and Half-Gates [ZRE15] garbled tables for the data-oblivious modular subtraction $[v - \eta]_{2^{b_{\mathrm{sh}}}}$. Using the standard one-AND-per-bit ripple-borrow recurrence $x_i = v_i \oplus \eta_i$, $\mathrm{diff}_i = x_i \oplus \mathrm{bor}_i$, and $\mathrm{bor}_{i+1} = \mathrm{bor}_i \oplus (x_i \wedge (\mathrm{bor}_i \oplus \eta_i))$ ($\mathrm{bor}_0 = 0$, with no borrow-out needed at $i = b_{\mathrm{sh}} - 1$), the circuit requires strictly $b_{\mathrm{sh}} - 1$ non-free AND gates for arbitrary secret inputs (v, η) with fixed topology. Extracting the k_{out} output wires immediately above the lowest $\log_2 \tau_{\mathrm{sh}}$ noise bits ($k_{\mathrm{out}} = k + 1$ for PH-PBS-GLWE refresh; $k_{\mathrm{out}} = k$ otherwise, omitting the zero sign bit in GLWE terminal decoding) yields active wires whose plaintext value equals μ (PH-PBS-Blind and terminal decoding), $\alpha = [\mu - r]_p$ (PH-PBS-LWE), or $z = [\mu - q_r]_{2p}$ (PH-PBS-GLWE) whenever the error bound holds. The garbled circuit thus scales only logarithmically in p and stays independent of d and M_v.

3.3 Single-LUT Refresh: PH-PBS-Blind

Under small-key LWE ($\mathbf{s} \in \{0,1\}^n$), PH-PBS-Blind binds blinded lookup directly to the k active output wire labels of the subtraction circuit and refreshes noise via fresh client re-encryption; it targets single-table refresh ($M_v = 1$) and extends directly to $M_v \ge 1$

parallel tables. This keeps the selected index hidden from $\mathcal{S}$ and hides both μ and every table output from $\mathcal{C}$. Let $\{L_w^0\}_{w=0}^{k-1}$ be the k output zero-labels of the subtraction circuit at node v, with point-and-permute mask $\pi \in \{0,\dots,p-1\}$ whose bits are $\pi_w = \mathrm{lsb}(L_w^0)$. Because XORing raw Free-XOR labels across output wires leaks Δ_{GC} relations, $\mathcal{S}$ hashes each output wire independently with a tweakable correlation-robust hash $H(x,\tau) = \pi_\tau(\mathsf{Lin}(x)) \oplus \mathsf{Lin}(x)$ (for a public invertible $\mathbb{F}_2$-linear orthomorphism Lin [GKW+20]) to obtain $U_{w,\iota} = H(L_w^0 \oplus \iota\Delta_{\mathrm{GC}}, \tau_{v,w})$ ($\iota \in \{0,1\}$), and forms $S_h = \bigoplus_{w=0}^{k-1} U_{w,h_w}$ for each index $h \in \{0,\dots,p-1\}$. Indexing the M_v tables by $t \in \{0,\dots,M_v-1\}$ in the row tweak ($M_v = 1$ in the primary single-table path), $\mathcal{S}$ samples a uniform blind $m_{\mathrm{blind},t} \in \mathbb{Z}_{2p}$ (chosen modulo $2p$ rather than p so homomorphic unblinding never flips the encoding padding bit) and, using a fresh per-invocation row permutation $\pi_{\mathrm{row},v}$, encrypts each entry at permuted slot $c = h \oplus \pi \in \{0,\dots,p-1\}$ as

$$X_{t,c} = \mathsf{Lin}(S_h) \oplus \big((t \ll 16) \,|\, c\big), \qquad \mathsf{Row}_{t,c} = \big[f_t(h) + m_{\mathrm{blind},t}\big]_{2p} \oplus \big[\pi_{\mathrm{row},v}(X_{t,c}) \oplus X_{t,c}\big]_{2^{k+1}}. \tag{4}$$

In Stage 3, $\mathcal{S}$ sends the packed $(k+1)$-bit rows $\{\mathsf{Row}_{t,c}\}$ alongside the subtraction garbled tables.

In Stage 4, $\mathcal{C}$ evaluates the subtraction circuit to obtain its k active output labels $\{\ell_w\}_{w=0}^{k-1}$ (encoding μ upon correct index extraction). From their selection bits alone, $\mathcal{C}$ identifies the unique active slot $c^* = (\mathrm{lsb}(\ell_0),\dots,\mathrm{lsb}(\ell_{k-1})) = \mu \oplus \pi$ without learning μ or π, computes $S_\mu = \bigoplus_{w=0}^{k-1} H(\ell_w, \tau_{v,w})$ and X_{t,c^*}, and decrypts only the active blinded entry $y'_t = \mathsf{Row}_{t,c^*} \oplus [\pi_{\mathrm{row},v}(X_{t,c^*}) \oplus X_{t,c^*}]_{2^{k+1}} = [f_t(\mu) + m_{\mathrm{blind},t}]_{2p}$ for each table. Because the small-key noise scales (Subsection 4.4) exceed the 32-bit grid step, $\mathcal{C}$ halves ciphertext body width by returning a fresh 20-byte seeded LWE encryption of y'_t on the 32-bit grid $\mathbb{Z}_{2^{32}}$ (128-bit mask seed, scale $\Delta_{32} = 2^{31-k}$, unquantized 32-bit body). Upon receipt, $\mathcal{S}$ expands the 32-bit mask from the seed, homomorphically subtracts $m_{\mathrm{blind},t}\Delta_{32} \pmod{2^{32}}$, and lifts both mask and body by 2^{32} to $\mathbb{Z}_q$. Because one-padding-bit encoding satisfies $2p \cdot \Delta_{32} = 2^{32} \equiv 0 \pmod{2^{32}}$, the modulo-$2p$ blind cancels identically without flipping the padding bit and produces a native modulus-q LWE ciphertext with fresh single-encryption noise encrypting $f_t(\mu)$ whenever index extraction succeeds.

While this wire-keyed mechanism handles $M_v \geq 1$ parallel tables on one input ciphertext (as summarized in full generality in Algorithm 1), both Stage 3 blinded-table traffic and Stage 4 re-encryption traffic grow linearly with M_v. When a node evaluates multiple private tables on the same input ciphertext, applying one input-independent, single-use token across all M_v tables at that node (and discarding it immediately afterward) removes this linear growth.

3.4 Multi-LUT Refresh: PH-PBS-LWE and PH-PBS-GLWE

PH-PBS-LWE and PH-PBS-GLWE hold online communication constant in M_v by having $\mathcal{C}$ prepare one input-independent, single-use encrypted shift token per refresh node and coalesce its payload into Stage 2 (Round 1) alongside the COT correction bits. Each token represents its mask compactly via a 128-bit seed ζ expanded by a public-seed PRG and transmits only p active body coefficients quantized to b_B bits; body quantization compresses Stage 2 bandwidth at a deterministic truncation error bounded in Section 4, with PH-PBS-LWE matching its small-key noise scale on $\mathbb{Z}_{2^{32}}$ and PH-PBS-GLWE operating directly on $\mathbb{Z}_q$ to preserve its smaller ring noise σ. Once $\mathcal{S}$ recovers the shift-masked discrete index α in Stage 4 (Round 2), it expands the token mask once, evaluates all M_v private tables $f_1,\dots,f_{M_v}$ locally with zero additional communication, and discards the token. All three refresh modes follow the unified 2-RTT four-half-round message flow in Algorithm 1.

PH-PBS-LWE (32-bit one-hot LWE token). The client samples a uniform shift $r \in \mathbb{Z}_p$ and a 128-bit seed ζ, expands p small-key masks $\mathbf{a}_0^{(32)},\dots,\mathbf{a}_{p-1}^{(32)} \in \mathbb{Z}_{2^{32}}^n$ via a public-seed

PRG, and encrypts the one-hot indicator $\Delta_{32}\cdot\mathbf{1}_{i=r}$ on $\mathbb{Z}_{2^{32}}$ as $b_i^{(32)} = [\langle \mathbf{a}_i^{(32)}, \mathbf{s}\rangle + \Delta_{32}\mathbf{1}_{i=r} + e_i^{(32)}]_{2^{32}}$ ($e_i^{(32)} \leftarrow \chi_\sigma$). Before sending the token in Stage 2, $\mathcal{C}$ quantizes the p bodies from the 32-bit source modulus via $\tilde{b}_i = \mathrm{MS}_{2^{32}\to 2^{b_B}}(b_i^{(32)})$ ($b_B \le 32$, lossless when $b_B = 32$) and transmits $(\zeta, \{\tilde{b}_i\}_{i=0}^{p-1})$. In Stage 4, $\mathcal{C}$ returns the k output selection bits, which $\mathcal{S}$ combines with its retained decoding bits to recover the shift-masked index α (equal to $[\mu - r]_p$ upon correct index extraction). Expanding the p masks once from ζ and lifting each quantized body to $\hat{b}_i^{(32)} = 2^{32-b_B}\tilde{b}_i$, $\mathcal{S}$ locally evaluates each private table f_t ($t \in \{1,\dots,M_v\}$) as

$$\mathbf{c}_t^{(32)} = \sum_{h=0}^{p-1} f_t(h)\cdot\big(\mathbf{a}_{[h-\alpha]_p}^{(32)},\, \hat{b}_{[h-\alpha]_p}^{(32)}\big) \pmod{2^{32}}, \tag{5}$$

and lifts $\mathbf{c}_t^{(32)}$ by 2^{32} to obtain a native modulus-q LWE ciphertext encrypting $f_t(\mu)$ whenever $\alpha = [\mu - r]_p$, since $[h-\alpha]_p = r$ holds if and only if $h = \mu$.

PH-PBS-GLWE (64-bit projected monomial token). In PH-PBS-GLWE, $\mathcal{C}$ operates directly over the 64-bit negacyclic ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N+1)$ ($N = 2048$, stride $\tau_{\mathrm{rot}} = N/p$) under polynomial secret $s(X) \in \mathcal{R}_q$. Sampling a uniform rotation shift $r = \tau_{\mathrm{rot}} q_r + r_{\mathrm{lo}} \in \mathbb{Z}_{2N}$ and a 128-bit seed ζ that expands via public-seed PRG G to the exact 64-bit mask polynomial $A = G(\zeta) \in \mathcal{R}_q$ (with zero mask quantization error), $\mathcal{C}$ targets the monomial ciphertext $C = (A, B)$ of phase $\Delta X^{-r} + E \pmod{X^N+1}$. Because each table polynomial $Q_{f_t}(X) = \sum_{h=0}^{p-1} f_t(h) X^{\tau_{\mathrm{rot}} h}$ places nonzero terms only at multiples of the stride $\tau_{\mathrm{rot}} = N/p$, only p of the N body coefficients of C can ever contribute to the constant term extracted by Ext_0. Specifically, define the active residue $\rho = [-r]_{\tau_{\mathrm{rot}}} \in \mathbb{Z}_{\tau_{\mathrm{rot}}}$ and the p-element active coordinate set $I_\rho = \{\rho + \tau_{\mathrm{rot}} h : 0 \le h < p\} \subset \{0,\dots,N-1\}$. Rather than generating all N body coefficients, $\mathcal{C}$ samples only p independent noise entries $\{e_i\}_{i\in I_\rho} \leftarrow \chi_\sigma^p$, computes the p active 64-bit body coefficients $B_i = [[X^i](As + \Delta X^{-r}) + e_i]_q$, quantizes them from the 64-bit source modulus as $\tilde{B}_i = \mathrm{MS}_{q\to 2^{b_B}}(B_i)$, and sends $(\zeta, \{\tilde{B}_i\}_{i\in I_\rho})$ in Stage 2. In Stage 4, $\mathcal{C}$ returns the $k+1$ output selection bits together with r_{lo}; $\mathcal{S}$ recovers the coarse index z and forms the shift-masked rotation index

$$\alpha = [r_{\mathrm{lo}} - \tau_{\mathrm{rot}} z]_{2N}, \tag{6}$$

which unconditionally satisfies $[-\alpha]_{\tau_{\mathrm{rot}}} = \rho$ and further satisfies $[\alpha - r]_{2N} = [-\tau_{\mathrm{rot}}\mu]_{2N}$ whenever $z = [\mu - q_r]_{2p}$ (since $2p\tau_{\mathrm{rot}} = 2N$). Expanding $A = G(\zeta)$ once and lifting $\hat{B}_i = 2^{64-b_B}\tilde{B}_i$ ($i \in I_\rho$), $\mathcal{S}$ executes projected evaluation $\mathsf{ProjEval}$ for each table f_t: it extracts the LWE mask from $Q_{f_t}(X)X^\alpha A(X) \bmod (X^N+1)$ via Ext_0 and computes the scalar LWE body directly from the p active coordinates as

$$b_{\mathrm{ext}}^{(t)} = \left[\sum_{h=0}^{p-1} (-1)^{\kappa_h} f_t(h)\hat{B}_{i_h}\right]_q, \quad i_h = [-\alpha - \tau_{\mathrm{rot}} h]_N \in I_\rho, \quad \kappa_h = \frac{\alpha + \tau_{\mathrm{rot}} h + i_h}{N}. \tag{7}$$

The output is directly a native N-dimensional LWE ciphertext under $\mathbf{s}_{\mathrm{ext}}$. The next two results establish the exact equivalence of projected evaluation and on-demand noise sampling.

Proposition 1 (Exact coefficientwise projected equivalence)**.** *Let $\hat{B}(X) \in \mathcal{R}_q$ be any polynomial agreeing with $\{\hat{B}_i\}_{i\in I_\rho}$ on I_ρ (with arbitrary coefficients outside I_ρ), and set $\hat{C} = (A, \hat{B})$. For every lookup table $f : \mathcal{M} \to \mathcal{M}$ and every $\alpha \in \mathbb{Z}_{2N}$ satisfying $[-\alpha]_{\tau_{\mathrm{rot}}} = \rho$, the N-dimensional LWE ciphertext produced by* $\mathsf{ProjEval}$ *equals $\mathrm{Ext}_0(Q_f(X)X^\alpha\hat{C})$ identically in all $N+1$ coordinates. Moreover, whenever $[\alpha - r]_{2N} = [-\tau_{\mathrm{rot}}\mu]_{2N}$ for $\mu \in \mathcal{M}$, its decryption phase under $\mathbf{s}_{\mathrm{ext}}$ satisfies*

$$\mathrm{Phase}_{\mathbf{s}_{\mathrm{ext}}}(\mathsf{ProjEval}) = \left[\Delta f(\mu) + \sum_{h=0}^{p-1} (-1)^{\kappa_h} f(h)\big(e_{i_h} + \delta_{B,i_h}\big)\right]_q, \tag{8}$$

where $\delta_{B,i} = \mathrm{ctr}_q(\hat{B}_i - B_i)$ *denotes the body quantization error at active coordinate* $i \in I_\rho$.

Proof. By definition, ProjEval computes the mask vector by applying Ext_0 to the product $Q_f X^\alpha A$ in $\mathcal{R}_q$, which matches the mask of $\mathrm{Ext}_0(Q_f X^\alpha \hat{C})$ coordinatewise. For the body, the constant term $[X^0](Q_f X^\alpha \hat{B})$ in $\mathcal{R}_q$ receives a contribution from $f(h)X^{\alpha+\tau_{\mathrm{rot}}h}$ and $\hat{B}_i X^i$ if and only if $\alpha + \tau_{\mathrm{rot}}h + i \equiv 0 \pmod N$. The unique index in $\{0,\dots,N-1\}$ is $i_h = [-\alpha-\tau_{\mathrm{rot}}h]_N$, and reduction modulo X^N+1 attaches the negacyclic sign $(-1)^{\kappa_h}$ with $\kappa_h = (\alpha+\tau_{\mathrm{rot}}h+i_h)/N \in \mathbb{Z}_{\geq 0}$. Because $\tau_{\mathrm{rot}} \mid N$, every i_h satisfies $i_h \equiv -\alpha \equiv \rho \pmod{\tau_{\mathrm{rot}}}$ and therefore lies in I_ρ; all $N-p$ inactive body coefficients drop out identically, so $b_{\mathrm{ext}} = [X^0](Q_f X^\alpha \hat{B})$. By the sample-extraction identity of Subsection 2.2, the decryption phase equals $[X^0](Q_f X^\alpha(\hat{B} - As)) \pmod q$. Substituting $\hat{B}_{i_h} - [As]_{i_h} \equiv [\Delta X^{-r}]_{i_h} + e_{i_h} + \delta_{B,i_h} \pmod q$ on I_ρ gives

$$[X^0]\big(Q_f(X)X^{\alpha-r}\Delta\big) + \sum_{h=0}^{p-1}(-1)^{\kappa_h} f(h)\big(e_{i_h} + \delta_{B,i_h}\big) \pmod q.$$

When $[\alpha - r]_{2N} = [-\tau_{\mathrm{rot}}\mu]_{2N}$, the relation $X^{2N} = 1$ in $\mathcal{R}_q$ implies that $Q_f X^{\alpha-r}\Delta$ equals $\sum_{h=0}^{p-1} f(h)\Delta X^{\tau_{\mathrm{rot}}(h-\mu)}$. Because $h, \mu \in \{0,\dots,p-1\}$, every $h \neq \mu$ satisfies $0 < \tau_{\mathrm{rot}}|h-\mu| \leq N - \tau_{\mathrm{rot}} < N$, so only $h = \mu$ contributes to the constant term, with sign $+1$, and recovers $\Delta f(\mu)$. □

Lemma 1 (On-demand active-coordinate noise equivalence)**.** *Fix* $(s(X), \zeta, r)$*, which fixes* $A = G(\zeta)$ *and* I_ρ*. Let* Full *sample full-length noise* $(e_0,\dots,e_{N-1}) \leftarrow \chi_\sigma^N$ *and retain only* $\{e_i\}_{i\in I_\rho}$ *to build* $\{\tilde{B}_i\}_{i\in I_\rho}$*, and let* Active *sample only* $\{e_i\}_{i\in I_\rho} \leftarrow \chi_\sigma^p$ *on* I_ρ*. Then both experiments induce strictly identical distributions on the transmitted payload* $(\zeta, \{\tilde{B}_i\}_{i\in I_\rho})$ *and on the* ProjEval *output ciphertext for any lookup table* f*.*

Proof. In Full, the N noise coordinates are independent and identically distributed from χ_σ and independent of $(s(X), \zeta, r)$, so their marginal distribution on the fixed p-element subset I_ρ equals χ_σ^p. By Proposition 1, both the transmitted quantized bodies and the ProjEval output ciphertext are deterministic functions of $(A, s(X), r, \alpha, f, \{e_i\}_{i\in I_\rho})$ and are independent of $\{e_i\}_{i\notin I_\rho}$, so the two distributions coincide identically. □

3.5 1.5-RTT Terminal Plaintext Decoding

For a designated terminal ciphertext $\mathbf{c}_{\mathrm{end}}$ (a single handle or pre-terminal affine combination within budgets (W_1, W_2) specified by $\mathcal{G}$), the parties reuse the CtrMS, PhasePads, and garbled-subtraction steps of Algorithm 1 with zero shift ($v_{\mathrm{end}} = [u_{\mathrm{end}} + \tau_{\mathrm{sh}}/2]_{2^{b_{\mathrm{sh}}}}$) and no token payload. In Stage 3, $\mathcal{S}$ sends the garbled subtraction tables together with the k decoding bits $\{d_w\}$ for the plaintext output wires immediately above the $\log_2 \tau_{\mathrm{sh}}$ noise bits (omitting the zero sign wire under PH-PBS-GLWE's half-domain contract), and omits Stage 4. Whenever at least b_{sh} COT instances are available, decoding finishes in 1.5 RTT (three half-rounds) and $\mathcal{C}$ recovers only the rounded terminal plaintext $y \in \mathcal{M}$ via $y_w = \mathrm{lsb}(\ell_w) \oplus d_w$ without observing $\mathbf{c}_{\mathrm{end}}$ or its unrounded phase, while $\mathcal{S}$ learns nothing about y (matching the target output under Section 4 and proved simulation-secure in Section 5).

4 Correctness Analysis and Parameter Certificates

In standard non-interactive TFHE, refresh noise stems from blind-rotation external products, gadget decompositions, and key switching [CGGI20, dRDV26]. In PH-PBS, discrete index selection at each node depends solely on the aggregated input phase error and

Algorithm 1 Unified 2-RTT refresh message flow of the PH-PBS suite (Round 1: Stages 1–2; Round 2: Stages 3–4).

Inputs. $\mathcal{S}$ holds a native modulus-q LWE ciphertext $\mathbf{c} = (\mathbf{a}, b)$ encrypting $\mu \in \mathcal{M}$ and M_v private LUTs $\{f_t\}_{t=1}^{M_v}$; $\mathcal{C}$ holds the mode secret key; both parties hold initialized PhasePads and COT/GC state (including the shared PRG seed for server circuit inputs).

Steady-state precondition. At least b_{sh} precomputed COT instances are available and, in PH-PBS-LWE/GLWE, one unused token is ready. If the COT pool is insufficient, both parties execute an inline OT-extension refill before Stage 1.

Outputs. $\mathcal{S}$ obtains M_v native modulus-q LWE ciphertexts $\{\mathbf{c}_{\mathrm{out},t}\}_{t=1}^{M_v}$ targeting $\{f_t(\mu)\}_{t=1}^{M_v}$ (exact upon correct index extraction; Section 4).

1. **Stage 1 (Round 1, $\mathcal{S} \to \mathcal{C}$, phase sharing):** $\mathcal{S}$ applies CtrMS via (1) and truncates to b_{sh} bits, invokes PhasePads via (2) to sample $\eta \in \mathbb{Z}_{2^{b_{\mathrm{sh}}}}$, and sends $(\gamma, \boldsymbol{\xi})$ to $\mathcal{C}$.
2. **Stage 2 (Round 1, $\mathcal{C} \to \mathcal{S}$, GC input and coalesced token):** $\mathcal{C}$ recovers u satisfying (3), forms the shifted subtraction input v, and consumes b_{sh} COT tuples (c_w^*, T_w) to retain active labels for v and form choice corrections $\mathsf{corr}_w = v_w \oplus c_w^*$.
   - *PH-PBS-Blind:* $\mathcal{C}$ sends corr.
   - *PH-PBS-LWE / PH-PBS-GLWE:* $\mathcal{C}$ sends the single-use token payload $((\zeta, \{\tilde{b}_i\}_{i=0}^{p-1})$ or $(\zeta, \{\tilde{B}_i\}_{i \in I_\rho}))$ coalesced with corr.
3. **Stage 3 (Round 2, $\mathcal{S} \to \mathcal{C}$, short garbled subtraction):** $\mathcal{S}$ derives η's input labels from the shared PRG seed and garbles $[v - \eta]_{2^{b_{\mathrm{sh}}}}$ into garbled tables GT ($b_{\mathrm{sh}} - 1$ non-free AND gates).
   - *PH-PBS-Blind:* $\mathcal{S}$ sends GT together with M_v point-and-permuted $\mathbb{Z}_{2p}$-blinded tables encrypted via (4).
   - *PH-PBS-LWE / PH-PBS-GLWE:* $\mathcal{S}$ sends GT and retains the k_{out} output decoding bits $\{d_w\}$.
4. **Stage 4 (Round 2, $\mathcal{C} \to \mathcal{S}$, response and local finalization):** $\mathcal{C}$ evaluates GT to obtain the k_{out} active output labels above the $\log_2 \tau_{\mathrm{sh}}$ noise bits.
   - *PH-PBS-Blind:* $\mathcal{C}$ decrypts only its active blinded slot c^* per table and returns M_v fresh seeded $\mathbb{Z}_{2^{32}}$ LWE ciphertexts; $\mathcal{S}$ expands each mask, unblinds by $m_{\mathrm{blind},t}\Delta_{32}$, and lifts to $\mathbb{Z}_q$ (Subsection 3.3).
   - *PH-PBS-LWE:* $\mathcal{C}$ returns the k output selection bits; $\mathcal{S}$ recovers α, evaluates all M_v LUTs via (5), lifts to $\mathbb{Z}_q$, and discards the token.
   - *PH-PBS-GLWE:* $\mathcal{C}$ returns the $k+1$ output selection bits and r_{lo}; $\mathcal{S}$ recovers z, sets α via (6), evaluates all M_v LUTs via $\mathsf{ProjEval}$ in (7), and discards the token.

centered modulus switching (CtrMS) rounding, and upon correct selection only the fresh output error generated by the current refresh enters downstream layers.

4.1 Exact Integer Phase Identity for Centered Modulus Switching

Whether the short subtraction circuit of Subsection 3.2 extracts the exact target index depends on the pre-truncation discrete phase $j_{\mathrm{ms}} = [\hat{b} - \langle \hat{\mathbf{a}}, \mathbf{s} \rangle]_{R_{\mathrm{sh}}}$; the mode-specific shared phase j is then obtained by truncation ($j = j_{\mathrm{ms}}$ for PH-PBS-GLWE, and $j = [j_{\mathrm{ms}}]_{p\tau_{\mathrm{sh}}}$ for PH-PBS-Blind and PH-PBS-LWE). For any refresh or terminal node v, let $\tilde{\mu}_v = \beta_v + \sum_{h \in \mathrm{Pred}(v)} w_{v,h} m_h \in \mathbb{Z}$ denote the unreduced integer affine sum (or $\tilde{\mu}_v \in \mathcal{M}$ for a fresh input ciphertext), and let $e_{\mathrm{in}} \in \mathbb{Z}$ be the integer error representative satisfying $\mathrm{Phase}_{\mathbf{s}}(\mathbf{c}_v^{\mathrm{in}}) \equiv \Delta\tilde{\mu}_v + e_{\mathrm{in}} \pmod q$ for the native modulus-q input ciphertext $\mathbf{c}_v^{\mathrm{in}} = (\mathbf{a}, b) \in \mathbb{Z}_q^d \times \mathbb{Z}_q$ (under $\mathbf{s}_{\mathrm{ext}}$ in PH-PBS-GLWE). In the error-propagation analysis below, e_{in} is

taken directly as the weighted integer linear combination of predecessor errors. The target plaintext index is $\mu_v = [\tilde{\mu}_v]_p$ in PH-PBS-Blind and PH-PBS-LWE, whereas PH-PBS-GLWE enforces the half-domain contract $0 \le \tilde{\mu}_v < p$, in which case $\mu_v = \tilde{\mu}_v$.

Recall from Subsection 3.2 that CtrMS targets modulus $R_{\mathrm{sh}} = 2^{b_{\mathrm{ms}}} = 2p \cdot \tau_{\mathrm{sh}}$ with step size $h_{\mathrm{ms}} = q/R_{\mathrm{sh}}$, mask residuals $\varepsilon_i = \mathrm{ctr}_q(h_{\mathrm{ms}}\hat{a}_i - a_i)$, and $\nu_i = \mathrm{trunc}(\varepsilon_i/2)$. While mean compensation [dRDV26] motivates canceling the bias of binary secrets, our protocol constructs $\delta_{\mathrm{ms}}(\mathbf{a})$ in (1) purely over integers; its parity-truncation residual is

$$\omega(\mathbf{a}) = \left(\sum_{i=0}^{d-1} \nu_i - \mathrm{trunc}\left(\frac{1}{2}\sum_{i=0}^{d-1}(2\nu_i - \varepsilon_i)\right)\right) - \frac{1}{2}\sum_{i=0}^{d-1}\varepsilon_i \in \left\{-\frac{1}{2},\, 0,\, +\frac{1}{2}\right\},$$

which yields the exact decomposition $\delta_{\mathrm{ms}}(\mathbf{a}) = \frac{1}{2}\sum_{i=0}^{d-1}\varepsilon_i - h_{\mathrm{ms}}/2 + \omega(\mathbf{a})$.

Lemma 2 (Centered selection and mode-truncation identity)**.** *Define the integer selection error $E_{\mathrm{sel}} = e_{\mathrm{in}} + \sum_{i=0}^{d-1}(\frac{1}{2} - s_i)\varepsilon_i + \omega(\mathbf{a}) \in \mathbb{Z}$ (an integer because $\frac{1}{2}\sum_{i=0}^{d-1}\varepsilon_i + \omega(\mathbf{a}) = \delta_{\mathrm{ms}}(\mathbf{a}) + h_{\mathrm{ms}}/2 \in \mathbb{Z}$). For every binary secret $\mathbf{s} \in \{0,1\}^d$, unreduced integer plaintext $\tilde{\mu}_v \in \mathbb{Z}$, and input ciphertext $\mathbf{c}_v^{\mathrm{in}}$, the pre-truncation discrete phase satisfies*

$$j_{\mathrm{ms}} = \left[\tau_{\mathrm{sh}}\tilde{\mu}_v + \left\lfloor \frac{E_{\mathrm{sel}}}{h_{\mathrm{ms}}} \right\rceil\right]_{R_{\mathrm{sh}}}. \tag{9}$$

For every even τ_{sh}, the discrete rounding error $\tilde{e} = \lfloor E_{\mathrm{sel}}/h_{\mathrm{ms}} \rceil$ lies in $[-\tau_{\mathrm{sh}}/2,\, \tau_{\mathrm{sh}}/2)$ if and only if

$$-\frac{\Delta}{2} \le E_{\mathrm{sel}} \le \frac{\Delta}{2} - 1. \tag{10}$$

Whenever (10) holds, the k_{out} output wires extracted immediately above the $\log_2 \tau_{\mathrm{sh}}$ noise bits in Subsection 3.2 satisfy:

1. *In PH-PBS-GLWE ($b_{\mathrm{sh}} = b_{\mathrm{ms}}$, $j = j_{\mathrm{ms}}$, $0 \le \tilde{\mu}_v < p$), they recover the coarse index $z = [\mu_v - q_r]_{2p}$.*
2. *In PH-PBS-Blind ($b_{\mathrm{sh}} = b_{\mathrm{ms}} - 1$, $j = [j_{\mathrm{ms}}]_{p\tau_{\mathrm{sh}}}$), the active output wire labels encode the modulo-p index $\mu_v = [\tilde{\mu}_v]_p$ without wraparound ambiguity (the client decrypts only the active blinded entry and does not learn μ_v).*
3. *In PH-PBS-LWE ($b_{\mathrm{sh}} = b_{\mathrm{ms}} - 1$, $j = [j_{\mathrm{ms}}]_{p\tau_{\mathrm{sh}}}$), they recover the shift-masked index $\alpha = [\mu_v - r]_p$.*
4. *In 1.5-RTT terminal decoding (zero shift), they deliver the target plaintext $y = \mu_v \in \mathcal{M}$ to the client.*

Proof. Using $a_i \equiv h_{\mathrm{ms}}\hat{a}_i - \varepsilon_i \pmod q$ and the decomposition of $\delta_{\mathrm{ms}}(\mathbf{a})$, we obtain

$$b + \delta_{\mathrm{ms}}(\mathbf{a}) - h_{\mathrm{ms}}\langle\hat{\mathbf{a}}, \mathbf{s}\rangle \equiv (b - \langle\mathbf{a}, \mathbf{s}\rangle) + \delta_{\mathrm{ms}}(\mathbf{a}) - \sum_{i=0}^{d-1} s_i\varepsilon_i \equiv \Delta\tilde{\mu}_v + E_{\mathrm{sel}} - \frac{h_{\mathrm{ms}}}{2} \pmod q.$$

Round-half-up body switching $\hat{b} = \lfloor([b + \delta_{\mathrm{ms}}(\mathbf{a})]_q + h_{\mathrm{ms}}/2)/h_{\mathrm{ms}}\rfloor \bmod R_{\mathrm{sh}}$ cancels the trailing $-h_{\mathrm{ms}}/2$; because $h_{\mathrm{ms}} \mid q$ and $\Delta = \tau_{\mathrm{sh}}h_{\mathrm{ms}}$, subtracting $\langle\hat{\mathbf{a}}, \mathbf{s}\rangle \bmod R_{\mathrm{sh}}$ yields (9). Since $\Delta/2 = (\tau_{\mathrm{sh}}/2)h_{\mathrm{ms}}$ is an integer, an integer E_{sel} satisfies $-\tau_{\mathrm{sh}}/2 \le \tilde{e} \le \tau_{\mathrm{sh}}/2 - 1$ if and only if (10) holds. Under (10), $\tilde{e} + \tau_{\mathrm{sh}}/2 \in \{0, \ldots, \tau_{\mathrm{sh}} - 1\}$. By (3), in PH-PBS-GLWE ($2^{b_{\mathrm{sh}}} = R_{\mathrm{sh}} = 2p\tau_{\mathrm{sh}}$ and $\tilde{\mu}_v = \mu_v$), the circuit computes $[v - \eta]_{R_{\mathrm{sh}}} = \tau_{\mathrm{sh}}[\mu_v - q_r]_{2p} + (\tilde{e} + \tau_{\mathrm{sh}}/2)$, so the $k+1$ wires above the low $\log_2 \tau_{\mathrm{sh}}$ bits recover $z = [\mu_v - q_r]_{2p}$. In PH-PBS-Blind and PH-PBS-LWE, $p\tau_{\mathrm{sh}} \mid R_{\mathrm{sh}}$ and $\tau_{\mathrm{sh}}\tilde{\mu}_v \equiv \tau_{\mathrm{sh}}\mu_v \pmod{p\tau_{\mathrm{sh}}}$, so truncation automatically removes all multiples of p and gives $j = [\tau_{\mathrm{sh}}\mu_v + \tilde{e}]_{p\tau_{\mathrm{sh}}}$. Hence

$[v-\eta]_{p\tau_{\mathrm{sh}}} = \tau_{\mathrm{sh}}[\mu_v - r]_p + (\tilde{e} + \tau_{\mathrm{sh}}/2)$, whose k high wires carry active labels encoding μ_v when $r = 0$ (PH-PBS-Blind) and recover $\alpha = [\mu_v - r]_p$ for shift r (PH-PBS-LWE). Terminal decoding sets zero shift, extracts the k wires immediately above the $\log_2 \tau_{\mathrm{sh}}$ noise bits (omitting the zero sign bit in PH-PBS-GLWE), and delivers $\mu_v \in \mathcal{M}$ to the client. □

A deterministic sufficient condition for correct selection at the current node is $|e_{\mathrm{in}}| + dh_{\mathrm{ms}}/4 + 1/2 < \Delta/2$. Indeed, $|1/2 - s_i| = 1/2$, $|\varepsilon_i| \le h_{\mathrm{ms}}/2$, and $|\omega(\mathbf{a})| \le 1/2$ imply $|E_{\mathrm{sel}}| < \Delta/2$, and because E_{sel} and $\Delta/2$ are integers, (10) holds without any distributional assumption on the mask.

Write the core selection error without parity truncation as $E'_{\mathrm{sel}} = e_{\mathrm{in}} + \sum_{i=0}^{d-1}(\frac{1}{2} - s_i)\varepsilon_i$, and define the normalized decoding margin $m_{\mathrm{dec}} = \Delta/(2q) = 1/(4p)$. Because $|\omega(\mathbf{a})| \le 1/2$ and the upper endpoint of (10) is tightened by 1 integer unit relative to $\Delta/2$, the more conservative symmetric sufficient condition

$$\frac{|E'_{\mathrm{sel}}|}{q} < m_{\mathrm{dec}} - \frac{1.5}{2^{64}} \tag{11}$$

strictly implies (10) and supports all three finite-task tail bounds below.

4.2 Output Errors and Shared-Token Reconvergence

To trace output noise into downstream DAG nodes, we represent each grid-rounded Gaussian sample on the native ciphertext grid $\mathbb{Z}_{q_{\mathrm{grid}}}$ ($q_{\mathrm{grid}} = 2^{32}$ lifted by 2^{32} to $\mathbb{Z}_q$ for PH-PBS-Blind and PH-PBS-LWE; $q_{\mathrm{grid}} = 2^{64}$ for PH-PBS-GLWE) in normalized form as $g + \delta_{\mathrm{rnd}}$, where $g \sim \mathcal{N}(0, \sigma^2)$ and $|\delta_{\mathrm{rnd}}| \le 1/(2q_{\mathrm{grid}})$. Token body quantization $\mathrm{MS}_{q_{\mathrm{grid}} \to 2^{b_B}}$ followed by lifting introduces normalized error δ_B/q bounded deterministically by $|\delta_B/q| \le \varepsilon_B$, with $\varepsilon_B = 0$ when no body quantization occurs (PH-PBS-Blind at all precisions and 8-bit PH-PBS-LWE at $b_B = 32$) and $\varepsilon_B = 2^{-(b_B+1)}$ otherwise.

After a correct refresh, the prior input error affects only discrete selection at the current node and does not propagate into the refreshed ciphertext; each output carries solely the fresh error produced by the current refresh. By Subsection 3.3, (5), and Proposition 1 ((8)), the normalized output phase error for table $f_t : \mathcal{M} \to \mathcal{M}$ takes two forms. PH-PBS-Blind returns a fresh encryption error $e_{\mathrm{out},t} = g_t + \delta_{\mathrm{rnd},t}$ ($\varepsilon_B = 0$, with Gaussian component of variance σ^2, independent of f_t). PH-PBS-LWE and PH-PBS-GLWE instead weight the p active token coordinates by the table entries:

$$e_{\mathrm{out},t} = \sum_{h=0}^{p-1} \varsigma_h f_t(h)\big(g_{\phi(h)} + \delta_{\mathrm{rnd},\phi(h)} + \delta_{B,\phi(h)}/q\big),$$

with $\varsigma_h = 1, \phi(h) = [h-\alpha]_p$ for PH-PBS-LWE and $\varsigma_h = (-1)^{\kappa_h}, \phi(h) = i_h \in I_\rho$ for PH-PBS-GLWE. Thus token noise scales with the LUT coefficients: every valid table satisfies $\|f_t\|_1 \le p(p-1)$ and $\|f_t\|_2^2 \le p(p-1)^2$, whereas a uniform random table satisfies $\mathbb{E}[\|f_t\|_2^2] = p(p-1)(2p-1)/6$; hence the worst-case to mean Gaussian squared-norm ratio is $\max_f \|f\|_2^2/\mathbb{E}[\|f\|_2^2] = 6(p-1)/(2p-1) < 3$. Under the uniform body-quantization variance approximation $\sigma_B^2 = 2^{-2b_B}/12$ ($\sigma_B^2 = 0$ when unquantized), the expected uniform-LUT variance $\mathbb{E}[\|f_t\|_2^2](\sigma^2 + \sigma_B^2)$ (or σ^2 for PH-PBS-Blind) serves as a single-refresh diagnostic reference against empirical sample variance in Subsection 6.2 (Table 4). All finite-task certificates below instead use worst-case LUT norms and deterministic quantization bounds with zero independence assumption on body quantization.

At a multi-input DAG node v, the server forms $\mathbf{c}_v^{\mathrm{in}} = \sum_{h \in \mathrm{Pred}(v)} w_{v,h}\mathbf{c}_h + (\mathbf{0}^d, [\Delta\beta_v]_q) \pmod q$. When parallel outputs evaluated from one multi-table token reconverge at node v, they share the same p underlying token noise coordinates, so their weighted errors add coherently rather than independently. Bounding aggregated variance therefore requires coalescing absolute weights within each encryption-noise source group $G \in \mathcal{P}_v$ (Subsection 2.4) before squaring.

Proposition 2 (Shared-token reconvergence error decomposition)**.** *Fix the task inputs, lookup tables, and affine weights, and define the error-free reference execution on the same underlying session randomness (keys, masks, encryption noises, and token shifts) by forcing every prior lookup node to select its prescribed ideal reference plaintext without conditioning on prior success; consider node v along this reference execution. Assume the local weights satisfy $\sum_{G\in\mathcal{P}_v}\overline{w}_{v,G}\le W_1$ and $\sum_{G\in\mathcal{P}_v}\overline{w}_{v,G}^2\le W_2^2$, with $\mathcal{P}_v$ and $\overline{w}_{v,G}$ as in Subsection 2.4. Set $(\Gamma_1,\Gamma_2)=(1,1)$ for PH-PBS-Blind and $(\Gamma_1,\Gamma_2)=(p(p-1),\,p(p-1)^2)$ for PH-PBS-LWE and PH-PBS-GLWE. Let sk denote the active binary secret ($\mathbf{s}$ in PH-PBS-Blind and PH-PBS-LWE; $\mathbf{s}_{\mathrm{ext}}$ in PH-PBS-GLWE) and let $\mathbf{r}_v$ collect the random shifts of all tokens contributing to node v (empty if none contribute). On the reference execution, write $e_{\mathrm{in}}^{(v)}/q=E_{\mathrm{gauss}}^{(v)}+E_{\mathrm{det}}^{(v)}$. Conditional on $(\mathsf{sk},\mathbf{r}_v)$, $E_{\mathrm{gauss}}^{(v)}$ is zero-mean Gaussian with conditional variance*

$$\sigma^2_{\mathrm{agg}}(\mathsf{sk},\mathbf{r}_v):=\mathrm{Var}\Big(E_{\mathrm{gauss}}^{(v)}\mid\mathsf{sk},\mathbf{r}_v\Big)\le W_2^2\cdot\Gamma_2\cdot\sigma^2, \tag{12}$$

and the combined grid-rounding and body-quantization shift satisfies (without any independence assumption on rounding or body quantization)

$$\left|E_{\mathrm{det}}^{(v)}\right|\le W_1\cdot\Gamma_1\cdot\left(\frac{1}{2q_{\mathrm{grid}}}+\varepsilon_B\right). \tag{13}$$

Proof. Fix $(\mathsf{sk},\mathbf{r}_v)$ on the reference execution. For an initial-input or PH-PBS-Blind group $G=\{h\}$, the single fresh Gaussian coordinate carries coefficient $w_{v,h}$, whose square is $\overline{w}_{v,G}^2\le\Gamma_2\overline{w}_{v,G}^2$ since $\Gamma_2\ge1$. For any PH-PBS-LWE or PH-PBS-GLWE token group G (including $|G|=1$), collect the coefficients multiplying each token noise coordinate across all handles $h=(v',t)\in G$: each handle contributes $\pm f_{v',t}(m)\,w_{v,h}$ for a unique LUT input $m\in\mathcal{M}$ with $f_{v',t}(m)\in\{0,\dots,p-1\}$, so the absolute coefficient on each coordinate is at most $(p-1)\overline{w}_{v,G}$. Across the p independent Gaussian coordinates of the token, the group's sum of squared coefficients is at most $p(p-1)^2\overline{w}_{v,G}^2=\Gamma_2\overline{w}_{v,G}^2$.

Because distinct groups in $\mathcal{P}_v$ use independent Gaussian draws under the single-use rule, conditional on $(\mathsf{sk},\mathbf{r}_v)$ the aggregate term $E_{\mathrm{gauss}}^{(v)}$ is a zero-mean Gaussian whose conditional variance satisfies

$$\mathrm{Var}\Big(E_{\mathrm{gauss}}^{(v)}\mid\mathsf{sk},\mathbf{r}_v\Big)\le\sigma^2\Gamma_2\sum_{G\in\mathcal{P}_v}\overline{w}_{v,G}^2\le W_2^2\cdot\Gamma_2\cdot\sigma^2,$$

which establishes (12). For the deterministic residual, each handle $h\in\mathrm{Pred}(v)$ contributes at most $|w_{v,h}|\Gamma_1(1/(2q_{\mathrm{grid}})+\varepsilon_B)$ in absolute value (including unquantized initial inputs since $\Gamma_1\ge1$ and $\varepsilon_B\ge0$); summing over all handles yields (13). □

4.3 Three-Tier Finite-Task Correctness Bounds

Combining Lemma 2 and Proposition 2, we establish three complementary non-asymptotic bounds on the first-divergence failure probability p_{fail} (the probability that at least one refresh or terminal node extracts an index differing from the error-free reference execution), under explicit modeling conditions:

Assumption A1 (Grid-rounded Gaussian model). Before grid rounding, all fresh encryption and token noise coordinates are mutually independent $\mathcal{N}(0,\sigma^2)$ draws, independent of the secret key sk, mask-generation randomness, and random token shifts; rounding to $\mathbb{Z}_{q_{\mathrm{grid}}}$ leaves a normalized residual satisfying $|\delta_{\mathrm{rnd}}|\le1/(2q_{\mathrm{grid}})$.

Assumption A2 (Key-conditioned CtrMS mask residues). Let $M_{\mathrm{ms}}=q_{\mathrm{grid}}/R_{\mathrm{sh}}$. At each refresh or terminal decoding on the reference execution, let A_v^{res} be the vector

of native-grid input mask coordinates reduced modulo M_{ms}. Conditional on the active binary secret sk, A_v^{res} is uniform on $\mathbb{Z}_{M_{\mathrm{ms}}}^d$ and independent of the aggregate Gaussian input term $E_{\mathrm{gauss}}^{(v)}$.

Assumption A3 (Deterministic body quantization). Token body quantization errors are bounded purely deterministically by $|\delta_B/q| \le \varepsilon_B$ with no independence or uniformity assumption.

Condition RK (Non-adaptive fixed task and random binary key). The inputs $\mathbf{x}$, lookup tables $\{f_{v,t}\}$, affine weights/biases $\{(w_{v,h}, \beta_v)\}$, and schedule $\mathcal{G}$ are fixed prior to session setup, independently of sk and protocol randomness; the secret bits of sk are sampled i.i.d. uniformly from $\{0,1\}$; and sk, mask-generation randomness, token shifts, and pre-rounding Gaussian noise are mutually independent.

Satisfying the weight budget (W_1, W_2) and satisfying Assumption A2 at every node are distinct conditions: an affine step such as $2\mathbf{c}$ or a token-mode LUT whose outputs are all even restricts output mask coordinates on $\mathbb{Z}_{q_{\mathrm{grid}}}$ to even residues, and 128-bit seed expansion does not by itself establish the information-theoretic mask uniformity required by Assumption A2 (the security of public-seed ciphertext distributions is formalized separately in Section 5). We therefore establish three certificates matched to explicit probability spaces and budgets: Theorem 1 bounds failure per fixed key under Assumptions A1–A3 (ideal mask residues) on the Table 2 budgets; Proposition 3 bounds failure over random key generation and execution under Condition RK and Assumptions A1 and A3 (with no distributional assumption on input masks) on the same Table 2 budgets; and Corollary 1 bounds failure per fixed key and arbitrary masks under Assumptions A1 and A3 (on the Table 2 budgets for PH-PBS-Blind and PH-PBS-LWE, and conservative budgets for PH-PBS-GLWE). Unlike classic TFHE analyses that approximate modulus-switching error as Gaussian and apply erfc [CGGI20, dRDV26], Theorem 1 accounts explicitly for the discrete endpoint mean bias via Jensen's inequality.

Theorem 1 (Conditional discrete Chernoff bound under Assumptions A1–A3)**.** *Consider a finite multi-input DAG task following public schedule $\mathcal{G}$ with L refresh nodes and N_{out} terminal decodings, where every node satisfies local budgets (W_1, W_2) (and $0 \le \tilde{\mu}_v < p$ in PH-PBS-GLWE). Under Assumptions A1–A3, set $a_{\mathrm{ms}} = 1/(4R_{\mathrm{sh}})$,*

$$\delta_{\mathrm{det}} = W_1\Gamma_1\left(\frac{1}{2q_{\mathrm{grid}}} + \varepsilon_B\right) + \frac{d}{4q_{\mathrm{grid}}} + \frac{1.5}{2^{64}}, \qquad m_{\mathrm{eff}} = m_{\mathrm{dec}} - \delta_{\mathrm{det}}.$$

Let $\mathcal{F}_v$ be the event that node v violates (11) on the error-free reference execution. If $m_{\mathrm{eff}} > 0$, then for every $\theta > 0$ and every refresh or terminal node v,

$$\Pr[\mathcal{F}_v \mid \mathsf{sk}] \le B_{\mathrm{node}}(\theta) := 2\exp\left(\frac{1}{2}\theta^2 W_2^2 \Gamma_2 \sigma^2 + d\ln\frac{\sinh(\theta a_{\mathrm{ms}})}{\theta a_{\mathrm{ms}}} - \theta m_{\mathrm{eff}}\right), \tag{14}$$

and for every fixed sk, the conditional first-divergence failure probability $p_{\mathrm{fail}|\mathsf{sk}}$ (and hence unconditionally p_{fail}) satisfies

$$p_{\mathrm{fail}|\mathsf{sk}} \le B_{\mathrm{task}} := (L + N_{\mathrm{out}}) \cdot B_{\mathrm{node}}^*, \qquad B_{\mathrm{node}}^* := \inf_{\theta > 0} B_{\mathrm{node}}(\theta). \tag{15}$$

Proof. Any first failure in the real execution implies that some refresh or terminal event $\mathcal{F}_v$ occurs on the reference execution. Fix such a node v. By Assumption A1 and Proposition 2, for every real t,

$$\mathbb{E}\left[\exp\left(tE_{\mathrm{gauss}}^{(v)}\right) \mid \mathsf{sk}, \mathbf{r}_v\right] = \exp\left(\frac{1}{2}t^2\sigma_{\mathrm{agg}}^2(\mathsf{sk}, \mathbf{r}_v)\right) \le \exp\left(\frac{1}{2}t^2 W_2^2 \Gamma_2 \sigma^2\right).$$

Because the right-hand side does not depend on $\mathbf{r}_v$, taking the conditional expectation over $\mathbf{r}_v$ given sk yields the same upper bound for $\mathbb{E}[\exp(tE^{(v)}_{\mathrm{gauss}}) \mid \mathsf{sk}]$.

Now condition solely on sk. By Assumption A2, the d CtrMS mask residues in A^{res}_v are mutually independent and uniform on $\mathbb{Z}_{M_{\mathrm{ms}}}$, and the vector A^{res}_v is independent of $E^{(v)}_{\mathrm{gauss}}$. Under round-half-up conversion $\mathrm{MS}_{q\to R_{\mathrm{sh}}}$ on $\mathbb{Z}_{q_{\mathrm{grid}}}$, each normalized residual ε_i/q is uniform on $\{-(M_{\mathrm{ms}}/2-1)/q_{\mathrm{grid}}, \ldots, (M_{\mathrm{ms}}/2)/q_{\mathrm{grid}}\}$ and therefore has conditional mean $1/(2q_{\mathrm{grid}})$; its contribution $(1/2 - s_i)/(2q_{\mathrm{grid}})$ to the conditional mean has absolute value at most $1/(4q_{\mathrm{grid}})$. After centering, $D_i = (1/2 - s_i)(\varepsilon_i/q - 1/(2q_{\mathrm{grid}}))$ is uniform on the M_{ms} equally spaced midpoints of $[-a_{\mathrm{ms}}, a_{\mathrm{ms}}]$ regardless of $s_i \in \{0,1\}$. Adding an independent zero-mean variable $V_i \sim \mathrm{Unif}[-a_{\mathrm{ms}}/M_{\mathrm{ms}},\, a_{\mathrm{ms}}/M_{\mathrm{ms}}]$ produces a continuous $\mathrm{Unif}[-a_{\mathrm{ms}}, a_{\mathrm{ms}}]$ variable, so Jensen's inequality ($\mathbb{E}[\exp(tV_i)] \geq 1$) gives

$$\mathbb{E}[\exp(tD_i) \mid \mathsf{sk}] \leq \frac{\sinh(ta_{\mathrm{ms}})}{ta_{\mathrm{ms}}} \qquad (t \neq 0).$$

Set $Z = E^{(v)}_{\mathrm{gauss}} + \sum_{i=0}^{d-1} D_i$. Conditional independence given sk yields $\mathbb{E}[\exp(tZ) \mid \mathsf{sk}] = \mathbb{E}[\exp(tE^{(v)}_{\mathrm{gauss}}) \mid \mathsf{sk}] \prod_{i=0}^{d-1} \mathbb{E}[\exp(tD_i) \mid \mathsf{sk}]$. Combining the deterministic residual bound (13), the total endpoint mean shift of magnitude at most $d/(4q_{\mathrm{grid}})$, and the integer allowance in (11) gives $\mathcal{F}_v \subseteq \{|Z| \geq m_{\mathrm{eff}}\}$. Applying the two one-sided Chernoff bounds at $t = \theta$ and $t = -\theta$ ($\theta > 0$) establishes (14) for every fixed sk; optimizing over $\theta > 0$ and applying a union bound over all $L + N_{\mathrm{out}}$ reference events yields (15). □

Proposition 3 (Random-key bound under Condition RK, A1, and A3)**.** *Consider the finite multi-input DAG task of Theorem 1 under Condition RK and Assumptions A1 and A3 (with no distributional assumption on input masks). Set*

$$\bar{\delta}_{\mathrm{det}} = W_1\Gamma_1\left(\frac{1}{2q_{\mathrm{grid}}} + \varepsilon_B\right) + \frac{1.5}{2^{64}}, \qquad \bar{m}_{\mathrm{eff}} = m_{\mathrm{dec}} - \bar{\delta}_{\mathrm{det}}.$$

If $\bar{m}_{\mathrm{eff}} > 0$, then over the joint randomness of key generation and protocol execution (including arbitrary even-scaled ciphertexts, all-even LUTs, and 128-bit PRG masks), the first-divergence failure probability satisfies

$$p_{\mathrm{fail}} \leq B^{\mathrm{rk}}_{\mathrm{task}} := 2(L + N_{\mathrm{out}}) \inf_{\theta>0} \exp\left(\frac{1}{2}\theta^2 W_2^2 \Gamma_2 \sigma^2 + d \ln\cosh(\theta a_{\mathrm{ms}}) - \theta\bar{m}_{\mathrm{eff}}\right). \tag{16}$$

Proof. Let $\mathcal{Z}$ denote all mask-generation randomness and token shifts across the session. Under Condition RK, along the error-free reference execution every node's reference plaintext μ_v is fixed by the task; in all three modes, initial masks, PH-PBS-Blind fresh masks, PH-PBS-LWE linear combinations $\sum_h f_t(h)\mathbf{a}^{(32)}_{[h-\alpha]_p}$, PH-PBS-GLWE extracted masks from $Q_{f_t}X^{\alpha}A$, and affine combinations of predecessor masks depend solely on the fixed task and $\mathcal{Z}$, and are independent of sk. Conditioned on $\mathcal{Z}$, every coordinate residual ε_i at node v is fixed with $|\varepsilon_i| \leq h_{\mathrm{ms}}/2 = q/(2R_{\mathrm{sh}})$, while the bits $s_i \in \{0,1\}$ of sk remain i.i.d. uniform and independent of $E^{(v)}_{\mathrm{gauss}}$. Hence $T_v = \sum_{i=0}^{d-1}(\frac{1}{2} - s_i)(\varepsilon_i/q)$ has exact conditional mean zero (requiring no endpoint bias subtraction) and satisfies

$$\mathbb{E}_{\mathsf{sk}}[\exp(\theta T_v) \mid \mathcal{Z}] = \prod_{i=0}^{d-1} \cosh\left(\frac{\theta\varepsilon_i}{2q}\right) \leq \cosh(\theta a_{\mathrm{ms}})^d$$

for every $\theta \in \mathbb{R}$, since cosh is even and increasing on $[0,\infty)$ and $|\varepsilon_i|/(2q) \leq a_{\mathrm{ms}}$. By Assumption A1 and Proposition 2, $E^{(v)}_{\mathrm{gauss}}$ is independent of sk given $\mathcal{Z}$ with MGF bounded by $\exp(\frac{1}{2}\theta^2 W_2^2\Gamma_2\sigma^2)$. Multiplying the two conditional MGFs, averaging over $\mathcal{Z}$, applying two-sided Chernoff bounds on $\{|E^{(v)}_{\mathrm{gauss}} + T_v| \geq \bar{m}_{\mathrm{eff}}\} \supseteq \mathcal{F}_v$ (which follows from (11) and

the pointwise deterministic bound (13), independently of any sk-dependence in δ_{rnd} or δ_B), optimizing over $\theta > 0$, and taking a union bound over all $L + N_{\mathrm{out}}$ nodes (which requires no independence across nodes under key reuse) yields (16). □

Corollary 1 (Fixed-key worst-case-mask bound under Assumptions A1 and A3)**.** *Consider the finite multi-input DAG task of Theorem 1 under Assumptions A1 and A3, for every fixed secret key* sk *and arbitrary input masks. Set*

$$\delta_{\mathrm{det}}^{\mathrm{wc}} = W_1 \Gamma_1 \left(\frac{1}{2q_{\mathrm{grid}}} + \varepsilon_B \right) + d a_{\mathrm{ms}} + \frac{1.5}{2^{64}}, \qquad m_{\mathrm{eff}}^{\mathrm{wc}} = m_{\mathrm{dec}} - \delta_{\mathrm{det}}^{\mathrm{wc}}.$$

If $m_{\mathrm{eff}}^{\mathrm{wc}} > 0$*, then for every fixed* sk *the first-divergence failure probability satisfies*

$$p_{\mathrm{fail}|\mathsf{sk}} \le B_{\mathrm{task}}^{\mathrm{wc}} := (L + N_{\mathrm{out}}) \cdot \mathrm{erfc}\left(\frac{m_{\mathrm{eff}}^{\mathrm{wc}}}{\sqrt{2}\, W_2 \sqrt{\Gamma_2}\, \sigma} \right). \tag{17}$$

Proof. Because $|1/2 - s_i| = 1/2$ and $|\varepsilon_i| \le h_{\mathrm{ms}}/2$ hold deterministically for every mask coordinate, $|q^{-1} \sum_{i=0}^{d-1} (\frac{1}{2} - s_i)\varepsilon_i| \le d h_{\mathrm{ms}}/(4q) = d a_{\mathrm{ms}}$. Combining this deterministic bound with (13) and (11) yields $\mathcal{F}_v \subseteq \{|E_{\mathrm{gauss}}^{(v)}| \ge m_{\mathrm{eff}}^{\mathrm{wc}}\}$. By Assumption A1 and Proposition 2, $E_{\mathrm{gauss}}^{(v)}$ is an exact zero-mean Gaussian conditional on $(\mathsf{sk}, \mathbf{r}_v)$ with standard deviation at most $W_2\sqrt{\Gamma_2}\,\sigma$, so $\Pr[\mathcal{F}_v \mid \mathsf{sk}, \mathbf{r}_v] \le \mathrm{erfc}(m_{\mathrm{eff}}^{\mathrm{wc}}/(\sqrt{2}\, W_2 \sqrt{\Gamma_2}\, \sigma))$ (holding trivially with 0 when $\sigma_{\mathrm{agg}} = 0$). Averaging over $\mathbf{r}_v$ and taking a union bound over all $L + N_{\mathrm{out}}$ nodes establishes (17). □

4.4 Parameter Selection and Numerical Results

Base lattice instances and native grids. Because PH-PBS eliminates blind rotation and key switching, it requires no BSK or KSK decomposition search and selects base lattice parameters directly for security and ciphertext compatibility. To target classical lattice security estimates above 132.0 bits (Section 5) without ad-hoc lattice instances, PH-PBS-Blind and PH-PBS-LWE adopt the official TFHE-rs V1_4 64-bit Gaussian 2^{-128} (PBS_KS small-key) LWE pairs (n, σ) across 2–8 bits, from $(914, 8.94 \times 10^{-7})$ to $(1267, 2.02 \times 10^{-9})$. PH-PBS-GLWE fixes the 4-bit rank-1 GLWE ring pair ($k_{\mathrm{glwe}} = 1, N = 2048, \sigma = 2.85 \times 10^{-15}$) across all four precisions; decoupling R_{sh} from N requires only $p \mid N$ ($\tau_{\mathrm{rot}} = 8$ at 8 bits) rather than expanding polynomial degree 32× to $N = 65536$ as in official 8-bit blind-rotation sets. The native ciphertext grid matches σ: in PH-PBS-Blind and PH-PBS-LWE, even the smallest $\sigma = 2.02 \times 10^{-9}$ exceeds the $\mathbb{Z}_{2^{32}}$ half-step $1.16 \times 10^{-10} \approx 0.058\sigma$ (rounding variance 0.11% of σ^2), so operating on $\mathbb{Z}_{2^{32}}$ with exact lifting to $\mathbb{Z}_q$ keeps fresh frames at 20 B and halves mask width; in PH-PBS-GLWE, $\sigma = 2.85 \times 10^{-15}$ lies over four orders of magnitude below the $\mathbb{Z}_{2^{32}}$ half-step and therefore runs directly on $\mathbb{Z}_q$ with zero mask quantization.

CtrMS oversampling and share widths. Under the continuous uniform envelope of Theorem 1, the CtrMS mask-rounding standard-deviation upper bound $\sigma_{\mathrm{ms}} = \sqrt{d/3}\, a_{\mathrm{ms}}$ (the exact discrete standard deviation carries factor $\sqrt{1 - M_{\mathrm{ms}}^{-2}} < 1$ under Assumption A2) satisfies the precision-invariant ratio $\sigma_{\mathrm{ms}}/m_{\mathrm{dec}} = \sqrt{d}/(2\sqrt{3}\,\tau_{\mathrm{sh}})$. Each additional bit in $\log_2 \tau_{\mathrm{sh}}$ adds one non-free AND gate (32 B garbled table and one COT) and roughly $(d+1)/8$ B of packed PhasePads payload while halving σ_{ms}. Setting $\tau_{\mathrm{sh}} = 2^{10}$ for PH-PBS-Blind and PH-PBS-LWE ($d \le 1267$) holds $\sigma_{\mathrm{ms}}/m_{\mathrm{dec}}$ within 0.85%–1.00%, and setting $\tau_{\mathrm{sh}} = 2^{11}$ for PH-PBS-GLWE ($d = 2048$) holds it at 0.64%; CtrMS rounding thus remains a minor fraction of the decoding margin across 2–8 bits with compact share widths $b_{\mathrm{sh}} \in [12, 20]$ (Table 2). Even under worst-case masks, this choice caps CtrMS mask rounding at $d a_{\mathrm{ms}}/m_{\mathrm{dec}} = d/(2\tau_{\mathrm{sh}}) \in [44.6\%, 61.9\%]$ (50.0% in PH-PBS-GLWE); this leaves 38.1%–55.4% of m_{dec} for $|e_{\mathrm{in}}|/q$ under Corollary 1.

Table 2: Parameters, certified weight budgets, and three-tier finite-task certificates across all 12 PH-PBS configurations ($L = 1024$ refreshes, $N_{\text{out}} = 64$ terminal decodings, worst-case LUTs; lattice estimates ≥ 132.0 bits in Section 5).

Mode	k (bit)	d	σ	$b_{\text{ms}}/b_{\text{sh}}$	b_B	(W_1, W_2)	Ideal ($\log_2 B_{\text{task}}$)	RK ($\log_2 B^{\text{rk}}_{\text{task}}$)	WC ($\log_2 \widehat{B}^{\text{wc}}_{\text{task}}$)
Blind	2	914	8.94×10^{-7}	13/12	32	(4096, 1024)	−2557.38	−1843.88	−1026.62
Blind	4	978	2.96×10^{-7}	15/14	32	(2048, 512)	−4646.93	−3049.58	−2084.80
Blind	6	1106	3.26×10^{-8}	17/16	32	(1024, 512)	−13554.11	−9476.17	−8378.25
Blind	8	1267	2.02×10^{-9}	19/18	32	(1024, 512)	−103661.83	−94353.70	−93096.51
LWE	2	914	8.94×10^{-7}	13/12	20	(256, 64)	−10522.06	−7627.91	−6721.83
LWE	4	978	2.96×10^{-7}	15/14	24	(64, 24)	−815.10	−689.48	−229.60
LWE	6	1106	3.26×10^{-8}	17/16	28	(16, 7)	−698.16	−591.22	−146.44
LWE	8	1267	2.02×10^{-9}	19/18	32	(4, 3)	−900.22	−714.38	−131.50
GLWE	2	2048	2.85×10^{-15}	14/14	20	(8192, 8192)	−1196.29	−375.41	$< -10^{9a}$
GLWE	4	2048	2.85×10^{-15}	16/16	24	(1536, 1536)	−1767.65	−545.69	$< -10^{9a}$
GLWE	6	2048	2.85×10^{-15}	18/18	28	(400, 400)	−1004.83	−316.41	$< -10^{9a}$
GLWE	8	2048	2.85×10^{-15}	20/20	40	(24576, 24576)	−1224.23	−384.67	-3698.77^{a}

Note: Blind/LWE use $\tau_{\text{sh}} = 2^{10}$, $q_{\text{grid}} = 2^{32}$ ($b_B = 32$ unquantized); GLWE uses $\tau_{\text{sh}} = 2^{11}$, $q_{\text{grid}} = 2^{64}$. Bounds are evaluated in log-domain from full-precision official TFHE-rs σ for Theorem 1 (Ideal, A1–A3), Proposition 3 (RK, Condition RK/A1/A3), and the Mills bound $\widehat{B}^{\text{wc}}_{\text{task}}$ on (17) (Corollary 1, A1/A3). [a]GLWE fixed-key WC budgets: $W_1 = W_2 \in \{5{,}120, 1{,}024, 256, 16{,}000\}$ at 2–8 bits (body quantization 46.9%–49.2% of m_{dec}).

Body quantization and weight budgets. In token modes, the body payload uses $\lceil p \cdot b_B/8 \rceil$ bytes, whereas Proposition 2 bounds the worst-case body-quantization fraction of margin m_{dec} by $\Delta_{\text{body}}/m_{\text{dec}} \leq 4pW_1\Gamma_1\varepsilon_B$, which equals $W_1 \cdot 2p^2(p-1) \cdot 2^{-b_B}$ when quantized ($\varepsilon_B = 2^{-(b_B+1)}$) and vanishes when $\varepsilon_B = 0$. Because $p^2(p-1)$ grows by roughly 64× per 2-bit precision increase, both token modes step b_B jointly against payload size and (14)–(16). In Gaussian-dominated PH-PBS-LWE, setting $b_B = 20, 24, 28$ at 2–6 bits keeps total deterministic shift within 2.3%–3.3% of m_{dec}, and $b_B = 32$ at 8 bits covers the full $\mathbb{Z}_{2^{32}}$ grid with zero body quantization (3.1% grid rounding, 1,040 B token). In quantization-dominated PH-PBS-GLWE (negligible σ), $b_B = 20, 24, 28$ matches the compact 26/64/240 B payloads at 2–6 bits so that worst-case body quantization uses 70.3%–76.9% of m_{dec} under the full Table 2 budgets; at 8 bits, adding one byte per entry to reach $b_B = 40$ (1,040 B to 1,296 B token) cuts body quantization 256× and lifts the full working budget from 96 to $W_1 = W_2 = 24{,}576$. Deployers can recalibrate (W_1, W_2) directly from (14)–(17) for target task sizes and bandwidth budgets.

Three-tier task certificates and budget interpretation. Evaluated in log-domain over the 1,088-decoding reference task ($L = 1024$, $N_{\text{out}} = 64$, worst-case LUTs), all 12 configurations stay below 2^{-698} under ideal mask residues (Theorem 1, A1–A3), below 2^{-316} over random keys and execution for fixed non-adaptive tasks (Proposition 3, Condition RK, A1, A3), and below 2^{-128} per fixed key under worst-case masks (Corollary 1, A1, A3; Table 2). Removing Assumption A2 leaves the listed budgets unchanged under Proposition 3 because the probability space averages over fresh random binary keys rather than conditioning on a fixed key. Under per-fixed-key worst-case masks (Corollary 1), PH-PBS-Blind and PH-PBS-LWE retain their listed budgets, whereas quantization-dominated PH-PBS-GLWE requires the conservative footnote budgets because worst-case CtrMS (50.0% of m_{dec}) plus the full body-quantization budgets (70.3%–76.9%) would otherwise exhaust the decoding margin. Deployers verify any multi-input affine node (including reconverging branches from a shared upstream token) by coalescing same-source weights via Proposition 2 against (W_1, W_2); Section 6 reports empirical single-refresh output phase variance diagnostics and validates multi-input DAG composition end-to-end across all three modes.

5 Security Analysis and Security Boundaries

We prove full-session simulation security in the static semi-honest model [Gol04] decoupled from specific noise models via the first-divergence failure probability p_{fail}, followed by concrete security estimates and deployment boundaries.

5.1 Security Model, Ideal Functionality, and Cryptographic Assumptions

Following the multi-input DAG model of Subsection 2.4, the client $\mathcal{C}$ holds private inputs $\mathbf{x} = (x_1, \ldots, x_{N_{\text{in}}}) \in \mathcal{M}^{N_{\text{in}}}$, while the server $\mathcal{S}$ holds private affine parameters $\{(w_{v,h}, \beta_v)\}$ within budgets (W_1, W_2) and private lookup tables $\{f_{v,t} : \mathcal{M} \to \mathcal{M}\}$ satisfying the mode's input-domain contract. The full-session ideal functionality $\mathcal{F}_{\text{PH-PBS}}$ is parameterized as follows:

- **Public state and allowed leakage.** The public execution schedule $\mathcal{G}$ specifies $(\text{mode}, k, d, L, N_{\text{out}})$, the refresh nodes $\{\text{Pred}(v), M_v, (W_1, W_2)\}_{v=1}^{L}$ (with total table evaluations $M_{\text{tot}} = \sum_{v=1}^{L} M_v$), and the N_{out} terminal decoding nodes $\{\text{Pred}_{\text{out}}(j), (W_1, W_2)\}_{j=1}^{N_{\text{out}}}$ (each identifying either a single handle or a pre-terminal affine combination within the certified weight budgets (W_1, W_2) and mode input-domain contract), together with the initial COT pool capacity C_{pool} and an exogenous stopping index $\tau_{\text{stop}} \in \{1, \ldots, L + N_{\text{out}}\} \cup \{\infty\}$ that models external transport interruption at invocation boundaries ($\tau_{\text{stop}} = \infty$ for a complete session). Because each invocation consumes b_{sh} COT instances and every stage's message length is fixed by (mode, k, d, M_v), all message sizes and inline COT refill points are deterministically fixed by $(\mathcal{G}, C_{\text{pool}})$.

- **Ideal outputs and truncation at invocation boundaries.** $\mathcal{F}_{\text{PH-PBS}}$ evaluates the DAG in topological order to compute each refresh node's reference plaintext μ_v, table outputs $m_{v,t} = f_{v,t}(\mu_v)$, and terminal plaintexts y_j on $\text{Pred}_{\text{out}}(j)$, delivers to $\mathcal{C}$ the prefix of terminal plaintexts $\mathbf{y}_{\le \tau_{\text{stop}}}$ for all terminal invocations completed up to τ_{stop} (the full vector $\mathbf{y} = (y_1, \ldots, y_{N_{\text{out}}}) \in \mathcal{M}^{N_{\text{out}}}$ when $\tau_{\text{stop}} = \infty$), and produces no application output for $\mathcal{S}$.

A static semi-honest adversary corrupts either $\mathcal{S}$ or $\mathcal{C}$ prior to execution and follows the protocol specification. Write $\mathsf{View}_{\mathcal{S}}^{\text{real}}$ and $\mathsf{View}_{\mathcal{C}}^{\text{real}}$ for the corrupted party's complete real-execution view. Simulation security requires probabilistic polynomial-time simulators $\text{Sim}_{\mathcal{S}}$ (given $\mathcal{G}$, C_{pool}, τ_{stop}, and $\{w_{v,h}, \beta_v, f_{v,t}\}$) and $\text{Sim}_{\mathcal{C}}$ (given $\mathcal{G}$, C_{pool}, τ_{stop}, $\mathbf{x}$, and $\mathbf{y}_{\le \tau_{\text{stop}}}$) whose outputs are computationally indistinguishable from the corresponding real views.

Computational indistinguishability rests on three cryptographic assumptions, while agreement between the real execution and the error-free reference execution enters generically through the first-divergence probability p_{fail} bounded in Section 4:

1. **Batched public-seed LWE and joint public-seed GLWE IND-CPA security.** We assume IND-CPA security for the seeded encryption distributions specified below, where every transmitted 128-bit mask seed ζ (for PH-PBS-Blind fresh ciphertexts and PH-PBS-LWE/GLWE single-use tokens) is part of the adversary's view; the games explicitly capture masks expanded from a common seed and all ciphertexts observed under the same encryption key:

    - *Batched public-seed LWE game (PH-PBS-Blind and PH-PBS-LWE):* The challenger samples one uniform binary secret $\mathbf{s} \leftarrow \{0,1\}^n$ and a hidden bit $\vartheta \leftarrow \{0,1\}$. On the j-th batch query ($j \in \{1, \ldots, Q_{\text{batch}}\}$), the adversary specifies

a batch length $\ell_j \geq 1$ and two equal-length message vectors $\mathbf{m}_{j,0}, \mathbf{m}_{j,1} \in \mathbb{Z}_{q_{\text{grid}}}^{\ell_j}$. The challenger samples a single fresh seed $\zeta_j \leftarrow \{0,1\}^\lambda$, expands ℓ_j masks $(\mathbf{a}_{j,0}, \ldots, \mathbf{a}_{j,\ell_j-1}) = G_{\ell_j}(\zeta_j) \in (\mathbb{Z}_{q_{\text{grid}}}^n)^{\ell_j}$, samples independent errors $e_{j,0}, \ldots, e_{j,\ell_j-1} \leftarrow \chi_\sigma$, and returns $(\zeta_j, \{[\langle \mathbf{a}_{j,i}, \mathbf{s}\rangle + m_{j,\vartheta,i} + e_{j,i}]_{q_{\text{grid}}}\}_{i=0}^{\ell_j-1})$. Single-ciphertext queries use $\ell_j = 1$ (initial ciphertexts uploaded with expanded masks are obtained by deterministic post-processing that expands ζ_j and drops the seed), whereas each PH-PBS-LWE token query uses $\ell_j = p$ (one shared seed generating p bodies). We write Q_{batch} for the number of seed batches, $Q_{\text{body}} = \sum_{j=1}^{Q_{\text{batch}}} \ell_j$ for the total body count, and $\mathsf{Adv}_{\text{pLWE}}(Q_{\text{batch}}, Q_{\text{body}})$ for the maximum distinguishing advantage.

- *Joint public-seed GLWE game (PH-PBS-GLWE):* The challenger samples one uniform binary coefficient vector $\mathbf{s}_{\text{ext}} = (s_0, \ldots, s_{N-1}) \leftarrow \{0,1\}^N$ (defining the binary-coefficient secret polynomial $s(X) = \sum_{i=0}^{N-1} s_i X^i \in \mathcal{R}_q$) and a hidden bit $\vartheta \leftarrow \{0,1\}$. On each of the N_{in} initial-ciphertext queries with message pair $(m_{j,0}, m_{j,1}) \in \mathbb{Z}_q^2$, the challenger encrypts $m_{j,\vartheta}$ under the common bit ϑ using the mode's mask expansion G under $s(X)$ and returns its expanded Ext_0 representation (an N-dimensional LWE ciphertext under $\mathbf{s}_{\text{ext}}$ by Subsection 2.2). On each of the L ring-token queries with message pair $(M_{v,0}, M_{v,1}) \in \mathcal{R}_q^2$, it samples a fresh seed $\zeta_v \leftarrow \{0,1\}^\lambda$ and $E_v \leftarrow \chi_\sigma^N$, and returns $(\zeta_v, G(\zeta_v)s + M_{v,\vartheta} + E_v)$. Body quantization, projection onto I_ρ, and modulus lifting are public deterministic post-processing. We denote the maximum distinguishing advantage by $\mathsf{Adv}_{\text{pGLWE}}(N_{\text{in}}, L)$.

Because our construction publishes neither BSK nor KSK, it requires no circular-security assumption for evaluation keys.

2. **PRF, miTCCR hashing, and ideal-permutation row masking.** We assume PRF security for the keyed PhasePads function. The Free-XOR and Half-Gates subtraction circuits [KS08, ZRE15] and the PH-PBS-Blind output-wire derivations share one hidden offset Δ_{GC} sampled uniformly from the odd λ-bit strings; we analyze all of their hash calls jointly in the multi-instance tweakable circular correlation-robust (miTCCR) game of Guo et al. [GKW+20, Definition 3] with distinct tweaks across the session. We model the row permutation $\pi_{\text{row},v}$ of each PH-PBS-Blind refresh node as an independent ideal permutation in a separate domain.

3. **Static semi-honest OT and COT extension.** We use Chou–Orlandi 1-out-of-2 base OT over P-256 [CO15] in static semi-honest composition together with IKNP OT extension (for PhasePads key delivery) and random-choice COT extension [IKNP03, ALSZ17], and denote their full-session semi-honest simulation advantages by $\epsilon_{\text{OT/COT},\mathcal{S}}$ and $\epsilon_{\text{OT/COT},\mathcal{C}}$.

5.2 Full-Session Simulation Security

Proving full-session simulation security for PH-PBS confronts two protocol-specific subtleties beyond single-invocation garbling or standalone encryption. In the corrupted client view, one global offset Δ_{GC} spans every subtraction circuit and every PH-PBS-Blind output-wire derivation across the session, while the p combined row keys $\{S_h\}$ in (4) satisfy deterministic $\mathbb{F}_2$-linear relations whenever $k \geq 2$ (such as $S_{00} \oplus S_{01} \oplus S_{10} \oplus S_{11} = 0$). In the corrupted server view, the server observes both each Stage 2 shift-encrypted token and the corresponding Stage 4 shift-masked index, and in PH-PBS-GLWE receives only the active projected body coordinates on I_ρ.

We first establish joint simulatability for the M_v packed blinded tables $\{\mathsf{Row}_{t,c}\}$ of PH-PBS-Blind in Subsection 3.3 ((4)) when all M_v tables at node v share one set of

active output wire labels $\{\ell_w\}_{w=0}^{k-1}$. Despite the $\mathbb{F}_2$-linear relations among $\{S_h\}$ for $k \geq 2$, the next lemma uses an explicit lazy-sampled ideal-permutation game to prove that pairwise uniformity of non-empty basis XORs suffices to replace all $M_v(p-1)$ inactive rows simultaneously against an adaptive query adversary.

Lemma 3 (Joint simulation of multi-table blinded payloads in PH-PBS-Blind)**.** *At a PH-PBS-Blind refresh node v with $n_{v,\mathrm{inact}} := M_v(p-1)$ inactive rows, suppose the visible transcript prior to the blinded tables is generated without reading Δ_{GC}, so that conditioned on this prefix and the client's active output wire labels $\{\ell_w\}_{w=0}^{k-1}$ (with zero-labels defined by $L_w^0 = \ell_w \oplus (\mu_{v,w}\Delta_{\mathrm{GC}})$, active slot $c^* = \mu_v \oplus \pi$, and active key S_{μ_v}), Δ_{GC} remains uniform over the $2^{\lambda-1}$ odd strings in $\{0,1\}^\lambda$. Model $\{\pi_{\tau_{v,w}}\}_{w=0}^{k-1}$ and $\pi_{\mathrm{row},v}$ as independent ideal permutations, and let an adaptive distinguisher make at most $Q_{\mathrm{wire},v,w}$ forward or inverse queries to $\pi_{\tau_{v,w}}$ (with $Q_{\mathrm{wire},v} = \sum_{w=0}^{k-1} Q_{\mathrm{wire},v,w}$) and at most $Q_{\mathrm{row},v}$ forward or inverse queries to $\pi_{\mathrm{row},v}$. Then the advantage of distinguishing the real M_v packed tables $\{\mathsf{Row}_{t,c}\}$ in (4) from simulated tables generated without μ_v, $\{f_{v,t}\}$, or Δ_{GC} (by sampling $y'_t \leftarrow \mathbb{Z}_{2p}$ independently for each t, setting $\mathsf{Row}_{t,c^*} = y'_t \oplus [\pi_{\mathrm{row},v}(X_{t,c^*}) \oplus X_{t,c^*}]_{2^{k+1}}$, and sampling $\mathsf{Row}_{t,c} \leftarrow \mathbb{Z}_{2p}$ independently and uniformly for all $c \neq c^*$) is at most*

$$\epsilon_{\mathrm{row},v} \leq \frac{3Q_{\mathrm{wire},v} + k + 2n_{v,\mathrm{inact}}Q_{\mathrm{row},v} + M_v^2\binom{p}{2} + \binom{n_{v,\mathrm{inact}}}{2} + n_{v,\mathrm{inact}}M_v}{2^\lambda}. \tag{18}$$

Proof. Define the simulated-table game $\mathsf{Game}_{\mathrm{sim}}$, in which the challenger computes the active blocks $U_{w,\mu_{v,w}}$ (one evaluation on each $\pi_{\tau_{v,w}}$), forms S_{μ_v} and the M_v active row inputs X_{t,c^*}, generates the active rows using independent uniform $y'_t \leftarrow \mathbb{Z}_{2p}$, samples all $n_{v,\mathrm{inact}}$ inactive rows $\mathsf{Row}_{t,c} \leftarrow \mathbb{Z}_{2p}$ ($c \neq c^*$) independently and uniformly without querying the permutations on inactive inputs, and answers all adversary forward and inverse permutation queries by lazy without-replacement sampling. In the real game, the independent uniform blind $m_{\mathrm{blind},t} \leftarrow \mathbb{Z}_{2p}$ makes $[f_{v,t}(\mu_v) + m_{\mathrm{blind},t}]_{2p}$ uniform in $\mathbb{Z}_{2p}$ and fixes $m_{\mathrm{blind},t} = [y'_t - f_{v,t}(\mu_v)]_{2p}$ once y'_t is fixed. After $\mathsf{Game}_{\mathrm{sim}}$ produces its visible transcript (the simulated tables, active labels, and lazy query-response pairs), sample Δ_{GC} uniformly over the $2^{\lambda-1}$ odd strings and sample $Y_w^* \leftarrow \{0,1\}^\lambda$ ($w \in \{0,\ldots,k-1\}$) independently and uniformly; this defines the k independent uniform basis differences $\Psi_w = U_{w,\mu_{v,w}} \oplus Y_w^* \oplus \mathsf{Lin}(\ell_w \oplus \Delta_{\mathrm{GC}}) \in \{0,1\}^\lambda$ and the hidden inactive inputs $X_{t,c} = \mathsf{Lin}(S_{\mu_v}) \oplus ((t \ll 16) \mid c) \oplus \bigoplus_{w:h_w \neq \mu_{v,w}} \mathsf{Lin}(\Psi_w)$ for $c \neq c^*$ ($h = c \oplus \pi \neq \mu_v$). Next, for each $c \neq c^*$, define the hidden output $Z_{t,c}^* \in \{0,1\}^\lambda$ by setting its low $k+1$ bits to $[X_{t,c}]_{2^{k+1}} \oplus \mathsf{Row}_{t,c} \oplus [f_{v,t}(h) + m_{\mathrm{blind},t}]_{2p}$ and sampling its upper $\lambda-k-1$ bits independently and uniformly from $\{0,1\}^{\lambda-k-1}$. Conditioned on any fixed transcript of $\mathsf{Game}_{\mathrm{sim}}$, each $X_{t,c}$ ($c \neq c^*$) contains a non-empty XOR of the independent uniform vectors $\{\mathsf{Lin}(\Psi_w)\}$ and is therefore marginally uniform over $\{0,1\}^\lambda$; because the low $k+1$ bits of $Z_{t,c}^*$ equal the low $k+1$ bits of $X_{t,c}$ XORed with a transcript-fixed offset while its upper $\lambda-k-1$ bits are independent and uniform, each $Z_{t,c}^*$ ($c \neq c^*$) is also marginally uniform over $\{0,1\}^\lambda$.

The real and simulated games are identical unless one of three bad events occurs in $\mathsf{Game}_{\mathrm{sim}}$ when programming $\pi_{\tau_{v,w}}(\mathsf{Lin}(\ell_w \oplus \Delta_{\mathrm{GC}})) = Y_w^*$ and $\pi_{\mathrm{row},v}(X_{t,c}) = Z_{t,c}^*$ ($c \neq c^*$):

1. *Wire-permutation collision ($\mathsf{Bad}_{\mathrm{wire}}$):* For some w, one of the $Q_{\mathrm{wire},v,w}$ queries to $\pi_{\tau_{v,w}}$ hits $\mathsf{Lin}(\ell_w) \oplus \mathsf{Lin}(\Delta_{\mathrm{GC}})$ (uniform over $2^{\lambda-1}$ values distinct from $\mathsf{Lin}(\ell_w)$ since Lin is a bijection and $\Delta_{\mathrm{GC}} \neq 0$, with probability at most $2Q_{\mathrm{wire},v,w}/2^\lambda$), or Y_w^* collides with one of the at most $Q_{\mathrm{wire},v,w}+1$ occupied outputs of $\pi_{\tau_{v,w}}$ (one active output plus $Q_{\mathrm{wire},v,w}$ query outputs, with probability at most $(Q_{\mathrm{wire},v,w}+1)/2^\lambda$). Summing over w gives $\Pr[\mathsf{Bad}_{\mathrm{wire}}] \leq (3Q_{\mathrm{wire},v}+k)/2^\lambda$.

2. *Row-permutation internal collision ($\mathsf{Bad}_{\mathrm{coll}}$):* Two distinct pairs $(t,c) \neq (t',c')$ share $X_{t,c} = X_{t',c'}$, or an inactive output $Z_{t,c}^*$ ($c \neq c^*$) collides with another inactive or active row output. For $c = c'$ and $t \neq t'$, $X_{t,c} \oplus X_{t',c} = (t \oplus t') \ll 16 \neq 0$. For $c \neq c'$

$(h \neq h')$, invertibility and $\mathbb{F}_2$-linearity of Lin make $\mathsf{Lin}(S_h \oplus S_{h'}) = \bigoplus_{w:h_w \neq h'_w} \mathsf{Lin}(\Psi_w)$ marginally uniform in $\{0,1\}^\lambda$, so $\Pr[X_{t,c} = X_{t',c'}] = 2^{-\lambda}$ across each of the $M_v^2\binom{p}{2}$ unordered pairs with $c \neq c'$. In the full execution (before fixing the transcript), the independent uniform rows $\mathsf{Row}_{t,c} \leftarrow \mathbb{Z}_{2p}$ $(c \neq c^*)$ and independent upper bits make the $n_{v,\mathrm{inact}}$ outputs $\{Z^*_{t,c}\}_{c \neq c^*}$ mutually independent and uniform in $\{0,1\}^\lambda$ and independent of the M_v active outputs, so output collisions across the $\binom{n_{v,\mathrm{inact}}}{2}$ inactive-inactive pairs and $n_{v,\mathrm{inact}}M_v$ inactive-active pairs occur with probability at most $(\binom{n_{v,\mathrm{inact}}}{2} + n_{v,\mathrm{inact}}M_v)/2^\lambda$.

3. *Row-permutation adaptive query hit ($\mathsf{Bad}_{\mathrm{hit}}$):* One of the adversary's $Q_{\mathrm{row},v}$ recorded forward or inverse queries (x, y) to $\pi_{\mathrm{row},v}$ satisfies $x = X_{t,c}$ or $y = Z^*_{t,c}$ for some $c \neq c^*$. Because $X_{t,c}$ and $Z^*_{t,c}$ each have uniform marginal distribution over $\{0,1\}^\lambda$ conditioned on the recorded transcript of $\mathsf{Game}_{\mathrm{sim}}$, a union bound over all $Q_{\mathrm{row},v}$ queries and $n_{v,\mathrm{inact}}$ inactive slots gives $\Pr[\mathsf{Bad}_{\mathrm{hit}}] \leq 2n_{v,\mathrm{inact}}Q_{\mathrm{row},v}/2^\lambda$.

Summing the three bounds yields (18). □

In the full-session reduction of Theorem 2, the miTCCR hybrid step already replaces all inactive wire blocks at node v with independent uniform strings, so only the row-permutation bad events ($\mathsf{Bad}_{\mathrm{coll}}$ and $\mathsf{Bad}_{\mathrm{hit}}$) of Lemma 3 remain; with $n_{v,\mathrm{inact}} = M_v(p-1)$, their combined cost is at most

$$\epsilon_{\mathrm{perm},v} = \frac{2n_{v,\mathrm{inact}}Q_{\mathrm{row},v} + M_v^2\binom{p}{2} + \binom{n_{v,\mathrm{inact}}}{2} + n_{v,\mathrm{inact}}M_v}{2^\lambda}.$$

Theorem 2 (Full-session static semi-honest simulation security)**.** *Consider a finite multi-input DAG session governed by public schedule $\mathcal{G}$, initial COT pool capacity C_{pool}, and stopping index τ_{stop} (N_{in} initial inputs, L refresh nodes, $M_{\mathrm{tot}} = \sum_{v=1}^{L} M_v$ total table evaluations, and N_{out} terminal decodings) satisfying the mode's input-domain contract. Let p_{fail} denote the probability that at least one completed refresh or terminal invocation up to τ_{stop} extracts a discrete index differing from the error-free reference execution. Under the cryptographic assumptions of Subsection 5.1, there exist probabilistic polynomial-time simulators $\mathrm{Sim}_{\mathcal{S}}$ and $\mathrm{Sim}_{\mathcal{C}}$ such that the distinguishing advantages for corrupted server and corrupted client views satisfy*

$$\epsilon_{\mathcal{S}} \leq \epsilon_{\mathrm{OT/COT},\mathcal{S}} + \mathsf{Adv}_{\mathrm{enc},\mathcal{S}} + p_{\mathrm{fail}}, \tag{19}$$

$$\epsilon_{\mathcal{C}} \leq \epsilon_{\mathrm{OT/COT},\mathcal{C}} + \epsilon_{\mathrm{sym},\mathcal{C}} + \epsilon_{\mathrm{row}} + p_{\mathrm{fail}}, \tag{20}$$

where $\mathsf{Adv}_{\mathrm{enc},\mathcal{S}} = \mathsf{Adv}_{\mathrm{pLWE}}(N_{\mathrm{in}} + M_{\mathrm{tot}}, N_{\mathrm{in}} + M_{\mathrm{tot}})$ and $\epsilon_{\mathrm{row}} = \sum_{v=1}^{L} \epsilon_{\mathrm{perm},v}$ in PH-PBS-Blind; $\mathsf{Adv}_{\mathrm{enc},\mathcal{S}} = \mathsf{Adv}_{\mathrm{pLWE}}(N_{\mathrm{in}} + L, N_{\mathrm{in}} + pL)$ and $\epsilon_{\mathrm{row}} = 0$ in PH-PBS-LWE; $\mathsf{Adv}_{\mathrm{enc},\mathcal{S}} = \mathsf{Adv}_{\mathrm{pGLWE}}(N_{\mathrm{in}}, L)$ and $\epsilon_{\mathrm{row}} = 0$ in PH-PBS-GLWE; and $\epsilon_{\mathrm{sym},\mathcal{C}}$ denotes the full-session PhasePads PRF and Half-Gates miTCCR hybrid losses (including the Lk output-wire derivations in PH-PBS-Blind). In particular, p_{fail} is bounded by B_{task} under Theorem 1 (Assumptions A1–A3), by $B^{\mathrm{rk}}_{\mathrm{task}}$ under Proposition 3 (Condition RK, A1, and A3), or by $B^{\mathrm{wc}}_{\mathrm{task}}$ per fixed key under Corollary 1 (Assumptions A1 and A3).

Proof. We construct the hybrid sequences and simulators for each corrupted party (truncating at invocation τ_{stop} whenever $\tau_{\mathrm{stop}} < \infty$).

Corrupted server view ($\mathrm{Sim}_{\mathcal{S}}$). Because $\mathcal{S}$ itself chooses the PhasePads PRF keys and the garbling offset Δ_{GC}, hiding the client's secret inputs relies solely on OT/COT sender privacy, single-use shift masking, and public-seed encryption security. Given $(\mathcal{G}, C_{\mathrm{pool}}, \tau_{\mathrm{stop}}, \{w_{v,h}, \beta_v, f_{v,t}\})$, we transition from the real execution H_0 to $\mathrm{Sim}_{\mathcal{S}}$'s output distribution H_4:

- $H_0 \to H_1$ *(OT/COT and choice-correction replacement, cost* $\epsilon_{\mathrm{OT/COT},\mathcal{S}}$*):* Replace the PhasePads OT extension and all session COT extensions with their semi-honest sender simulators (using neither $\mathbf{s}$ nor $s(X)$). In Stage 2 of each invocation, the client's b_{sh} COT choice-correction bits $\mathsf{corr}_w = v_w \oplus c_w^*$ are masked by single-use uniform receiver choice bits c_w^*; jointly reparameterizing the COT tuple by sampling $\mathsf{corr}_w \leftarrow \{0,1\}$ uniformly and setting $c_w^* = v_w \oplus \mathsf{corr}_w$ and $T_w = Q_w \oplus (c_w^* \Delta_{\mathrm{GC}})$ preserves the exact joint distribution of the COT state and Stage 2 message identically.
- $H_1 \to H_2$ *(Coupling to the error-free reference execution, cost* p_{fail}*):* In H_2, replace the output index computed by each subtraction circuit with the prescribed reference index of the error-free execution (for token-mode Stage 4 selection bits and PH-PBS-Blind Stage 4 re-encryption). Coupling H_1 and H_2 on identical session randomness, the two views coincide whenever no completed invocation up to τ_{stop} diverges from the reference execution, so their statistical distance is at most p_{fail}.
- $H_2 \to H_3$ *(Joint reparameterization of token shifts and Stage 4 masked indices, zero loss):* Terminal decoding and PH-PBS-Blind return no output selection bits. In token modes, $\mathcal{S}$ observes both the Stage 2 token (encrypting shift r) and the Stage 4 selection bits $\mathsf{sel}_w = y_w \oplus d_w$ (from which $\mathcal{S}$ recovers the shift-masked index and evaluates all M_v tables locally). Because each single-use token samples r independently and uniformly: in PH-PBS-LWE, $(\alpha, r) = ([\mu_v - r]_p, r)$ is identically distributed to first sampling uniform $\alpha \leftarrow \mathbb{Z}_p$ and setting $r = [\mu_v - \alpha]_p$; in PH-PBS-GLWE, uniform $r = \tau_{\mathrm{rot}} q_r + r_{\mathrm{lo}} \in \mathbb{Z}_{2N}$ is equivalent to independent uniform $(q_r, r_{\mathrm{lo}}) \in \mathbb{Z}_{2p} \times \mathbb{Z}_{\tau_{\mathrm{rot}}}$, so $(z, r_{\mathrm{lo}}, q_r) = ([\mu_v - q_r]_{2p}, r_{\mathrm{lo}}, q_r)$ is identically distributed to first sampling uniform $(z, r_{\mathrm{lo}}) \leftarrow \mathbb{Z}_{2p} \times \mathbb{Z}_{\tau_{\mathrm{rot}}}$ (which fixes $\alpha = [r_{\mathrm{lo}} - \tau_{\mathrm{rot}} z]_{2N}$, residue $\rho = [-\alpha]_{\tau_{\mathrm{rot}}}$, and active set I_ρ prior to Stage 2) and setting $q_r = [\mu_v - z]_{2p}$. Hybrid H_3 samples α (or (z, r_{lo})) first and programs the Stage 4 selection bits directly from $\mathcal{S}$'s decoding bits d_w as $\mathsf{sel}_w = \alpha_w \oplus d_w$ (or $z_w \oplus d_w$ together with r_{lo}).
- $H_3 \to H_4$ *(Batched public-seed zero replacement, cost* $\mathsf{Adv}_{\mathrm{enc},\mathcal{S}}$*):* In PH-PBS-Blind and PH-PBS-LWE, invoke the batched public-seed LWE game of Subsection 5.1 to replace the N_{in} initial ciphertexts ($\ell_j = 1$, uploaded with expanded masks) and either the M_{tot} Stage 4 fresh seeded ciphertexts in PH-PBS-Blind ($\ell_j = 1$) or the L Stage 2 shared-seed tokens in PH-PBS-LWE ($\ell_j = p$ bodies expanded from one seed ζ_v and quantized via $\mathrm{MS}_{2^{32} \to 2^{b_B}}$) with encryptions of all-zero message vectors under a fresh simulator secret key. In PH-PBS-GLWE, invoke the joint public-seed GLWE game to replace the N_{in} initial Ext_0 ciphertexts and L ring tokens with all-zero encryptions; by Lemma 1, transmitting the quantized active coordinates $(\zeta_v, \{\tilde{B}_i\}_{i \in I_\rho})$ on the active set I_ρ fixed in H_3 is identically distributed to public projection and quantization of a full public-seed GLWE challenge. In H_4, given the sampled public projection metadata, the encrypted message payloads of all ciphertexts and tokens are replaced with zeros and carry no information about r, μ_v, or $\mathbf{x}$.

The simulator $\mathrm{Sim}_{\mathcal{S}}$ executes H_4 by generating zero-message initial ciphertexts under a local key and sending uniform Stage 2 choice-correction bits at every refresh and terminal decoding. In PH-PBS-LWE it samples α uniformly, and in PH-PBS-GLWE it samples (z, r_{lo}) uniformly to fix I_ρ; in both token modes it sends the corresponding zero-message token in Stage 2 and returns the matching selection bits in Stage 4. In PH-PBS-Blind it returns M_v fresh zero-message seeded ciphertexts in Stage 4, and in all modes runs the server's prescribed local evaluation. This requires neither $\mathbf{x}$ nor $\mathbf{s}$ and establishes (19).

Corrupted client view ($\mathrm{Sim}_{\mathcal{C}}$). Given $(\mathcal{G}, C_{\mathrm{pool}}, \tau_{\mathrm{stop}}, \mathbf{x}, \mathbf{y}_{\le \tau_{\mathrm{stop}}})$, $\mathrm{Sim}_{\mathcal{C}}$ runs the corrupted client's initialization to sample and retain the client random tape and secret key $\mathbf{s}$ (or $s(X)$) consistent with the simulated execution, and transitions from G_0 to G_2:

- $G_0 \to G_1$ *(Joint simulation of PhasePads, subtraction garbling, and blinded tables, cost* $\epsilon_{\mathrm{OT/COT},\mathcal{C}} + \epsilon_{\mathrm{sym},\mathcal{C}} + \epsilon_{\mathrm{row}}$*):* First, replace the PhasePads OT extension and session COT extensions with their semi-honest receiver simulators (cost $\epsilon_{\mathrm{OT/COT},\mathcal{C}}$): the simulator generates the client's COT choice bits c_w^* and active input labels T_w directly without reading Δ_{GC} (leaving the server zero-labels defined implicitly by $Q_w = T_w \oplus (c_w^* \Delta_{\mathrm{GC}})$). Second, in Stage 1 of each invocation, replace the unselected PhasePads PRF outputs $\{P_{i,1-s_i}\}_{i=0}^{d-1}$ with uniform values (cost included in the PRF component of $\epsilon_{\mathrm{sym},\mathcal{C}}$); this makes $\xi_i = [P_{i,0} - P_{i,1} - \hat{a}_i]_{2^{b_{\mathrm{sh}}}}$ uniform conditional on (P_{i,s_i}, s_i), while $\mathcal{S}$'s fresh uniform share $\eta \leftarrow \mathbb{Z}_{2^{b_{\mathrm{sh}}}}$ makes $u = [j+\eta]_{2^{b_{\mathrm{sh}}}}$ uniform conditional on $\boldsymbol{\xi}$, so $\mathrm{Sim}_{\mathcal{C}}$ samples $\boldsymbol{\xi} \leftarrow \mathbb{Z}_{2^{b_{\mathrm{sh}}}}^d$ and $u \leftarrow \mathbb{Z}_{2^{b_{\mathrm{sh}}}}$ independently and sets $\gamma = [u - \sum_{i=0}^{d-1}(P_{i,s_i} + s_i\xi_i)]_{2^{b_{\mathrm{sh}}}}$ without reading Δ_{GC}. Third, we use a single reduction to the single-instance, single-query-per-tweak miTCCR oracle $\mathcal{O}^{\mathrm{miTCCR}}_{\Delta_{\mathrm{GC}}}(w,\tau,\mathrm{bit}) = H(w \oplus \Delta_{\mathrm{GC}}, \tau) \oplus \mathrm{bit} \cdot \Delta_{\mathrm{GC}}$ (with $H(x,\tau) = \pi_\tau(\mathsf{Lin}(x)) \oplus \mathsf{Lin}(x)$) of Guo et al. [GKW+20, Definition 3 and Theorem 2] to cover all $L + N_{\mathrm{out}}$ Half-Gates transcripts and all PH-PBS-Blind output-wire derivations jointly across the session. Starting from the COT-simulated active client input labels T_w and the active server input labels derived from the setup shared PRG seed (Subsection 3.2), the reduction constructs each half gate following the simulator in the proof of [GKW+20, Section 5] and propagates active wire labels bottom-up. At each PH-PBS-Blind refresh node $v \in \{1,\dots,L\}$ (where the circuit selects output index $\hat{\mu}_v \in \mathcal{M}$ with bits $\hat{\mu}_{v,w}$ and active output labels $\{\ell_w\}_{w=0}^{k-1}$), the reduction sets $\pi_w = \mathrm{lsb}(\ell_w) \oplus \hat{\mu}_{v,w}$, computes the active wire block publicly as $U_{w,\hat{\mu}_{v,w}} = H(\ell_w, \tau_{v,w})$, and obtains each inactive block by querying the same hidden-offset oracle on $(\ell_w, \tau_{v,w}, 0)$, which returns $H(\ell_w \oplus \Delta_{\mathrm{GC}}, \tau_{v,w})$ in the real game.

  Because every half gate (two per AND gate) and every PH-PBS-Blind output-wire derivation uses a distinct tweak, the reduction runs under one hidden offset Δ_{GC} and makes at most one hidden-offset query per tweak, as required by [GKW+20, Definition 3]. The total hidden-offset query count is at most $2(L + N_{\mathrm{out}})(b_{\mathrm{sh}} - 1)$ for the subtraction circuits, plus Lk in PH-PBS-Blind, and the joint oracle switch is charged once to $\epsilon_{\mathrm{sym},\mathcal{C}}$.

  In the ideal-oracle hybrid, the Half-Gates simulator samples uniform garbled table blocks (T_G, T_E) and obtains the active output labels $\{\ell_w\}$ by evaluating the resulting circuits without using internal wire truth values, while all inactive PH-PBS-Blind wire blocks $\{U_{w,1-\hat{\mu}_{v,w}}\}$ become independent uniform strings in $\{0,1\}^\lambda$. At each PH-PBS-Blind node v, the simulator therefore identifies the active slot $c^* = (\mathrm{lsb}(\ell_0), \dots, \mathrm{lsb}(\ell_{k-1}))$ directly from the active output labels, uses that slot across all M_v tables at node v, and samples an independent fresh uniform blinded value $y'_t \leftarrow \mathbb{Z}_{2p}$ for each table; applying the row-permutation step of Lemma 3 at $\hat{\mu}_v$ across $v = 1,\dots,L$ replaces all inactive rows without $\hat{\mu}_v$, $\{f_{v,t}\}$, or Δ_{GC} at additional cost at most $\epsilon_{\mathrm{row}} = \sum_{v=1}^{L} \epsilon_{\mathrm{perm},v}$.

- $G_1 \to G_2$ *(Coupling terminal decoding to prescribed ideal outputs, cost* p_{fail}*):* At the i-th terminal decoding ($i \in \{1,\dots,N_{\mathrm{out}}\}$), replace the real-execution terminal plaintext $\hat{y}_i$ with the ideal output y_i from $\mathbf{y}_{\le \tau_{\mathrm{stop}}}$ when setting the Stage 3 output decoding bits to $d_w = \mathrm{lsb}(\ell_w) \oplus y_{i,w}$, so $\mathcal{C}$ decodes y_i without observing any unrounded phase. Coupling G_1 and G_2 on identical session randomness, the two views coincide whenever no completed refresh or terminal decoding up to τ_{stop} diverges from the reference execution, so their statistical distance is at most p_{fail}.

Executing G_2 uses neither $\{(w_{v,h}, \beta_v)\}$ nor $\{f_{v,t}\}$ and establishes (20). □

5.3 Concrete Security Estimates

We report our concrete security metrics in two blocks (Table 3): estimated classical lattice attack costs for the five native-grid lattice instances (Block A), and accounted online-session symmetric/statistical query-and-collision terms $B_{\text{counted},\mathcal{S}}$ and $B_{\text{counted},\mathcal{C}}$ for reference refresh-and-terminal workloads (Block B).

For Block A, we evaluate the five underlying native-grid lattice instances using the Lattice Estimator [APS15] (commit 352ddaf4) under standard uniform-mask models with a binary secret distribution, the geometric series assumption (GSA), and unlimited samples, and record the minimum operation count $\log_2(\text{rop})$ across primal (uSVP), decoding (BDD), and dual-hybrid attack families under both MATZOV [MAT22] and BDGL16 [BDGL16]. In PH-PBS-Blind and PH-PBS-LWE, any modulus-q ciphertext lifted from $\mathbb{Z}_{2^{32}}$ has coefficients divisible by 2^{32} and can be publicly divided by 2^{32} back to the exact native $\mathbb{Z}_{2^{32}}$ sample, so lattice security is assessed directly on the native $q_{\text{grid}} = 2^{32}$ instances. All five instances yield estimated attack costs of 132.00–133.09 bits under MATZOV and 134.42–135.04 bits under BDGL16 (Table 3, Block A). Because Theorem 2 formally assumes the public-seed IND-CPA games $(\mathsf{Adv}_{\text{pLWE}}, \mathsf{Adv}_{\text{pGLWE}})$ of Subsection 5.1, Block A serves as an attack-cost estimate for the underlying lattice instances rather than a standalone reduction from standard uniform-mask LWE/GLWE to public-seed expansion.

For Block B, we fix an offline query budget $Q = 2^{40}$ over $L = 1024$ refreshes and $N_{\text{out}} = 64$ terminal decodings. For P-256 base OT, the generic-group ECDH/CDH term evaluates to $Q^2/2^{256} = 2^{-176.0}$ (full base-OT concrete security also depends on the chosen reduction, KDF model, and instance count). Following our implementation's budget calculator, the client accounted term $B_{\text{counted},\mathcal{C}} = (N'_{\text{tot}}Q + (N'_{\text{tot}})^2/2)/2^{128}$ sums query and collision terms over $N'_{\text{tot}} = (L + N_{\text{out}})(d + 2(b_{\text{sh}} - 1)) + N_{\text{blind}}$ tweak invocations ($N_{\text{blind}} = Lk + M_{\text{tot}}p$ for Blind; $N_{\text{blind}} = 0$ for token modes). In our implementation, initial input ciphertexts are uploaded with full expanded masks (using the same underlying mask-expansion routine as online ciphertexts/tokens and captured inside $\mathsf{Adv}_{\text{enc},\mathcal{S}}$ with N_{in} included), whereas online refreshes transmit $N^{\text{on}}_{\text{seed}} = M_{\text{tot}}$ public seeds in PH-PBS-Blind or $N^{\text{on}}_{\text{seed}} = L$ in PH-PBS-LWE/GLWE (terminal decoding transmits no mask seed). Under an ideal-cipher heuristic accounting model for AES-128-CTR seed expansion (B_{blk} 128-bit blocks per seed), the server accounted term $B_{\text{counted},\mathcal{S}} = (N_{\text{tot}}Q + N^2_{\text{tot}}/2)/2^{128} + N^{\text{on}}_{\text{seed}}B^2_{\text{blk}}/2^{129}$ conservatively combines this CTR switching term with $N_{\text{tot}} = (L + N_{\text{out}})(d + 2(b_{\text{sh}} - 1)) + N^{\text{on}}_{\text{seed}}$ online symmetric calls; we present this as explicit per-session heuristic accounting rather than a full reduction lemma, while private noise seeds and OT-extension streams remain covered under their respective PRG and COT assumptions.

Across both 1-LUT/node and 16-LUT/node reference sessions ($Q = 2^{40}$), the counted online terms $B_{\text{counted},\mathcal{S}}$ and $B_{\text{counted},\mathcal{C}}$ stay between $2^{-68.04}$ and $2^{-65.58}$ across all 12 configurations and remain invariant for PH-PBS-LWE and PH-PBS-GLWE as fan-out scales to 16 (Table 3, Block B).

5.4 Security Boundaries

Our security guarantees hold against static semi-honest corruption; extending PH-PBS to an active adversary that deviates arbitrarily from the protocol requires verifying three classes of interactive objects to prevent additional leakage: (1) consistency of the server's subtraction circuit and blinded tables with the prescribed functionality (preventing a modified circuit from exposing low-order phase residuals or inconsistent table offsets); (2) consistency between the client's Stage 2 single-use token (including its shift r) and garbled-circuit input v; and (3) authenticity of the client's Stage 4 output selection bits or fresh re-encryptions. Actively secure base OT and COT extension protect only the OT transfers, so active security additionally requires authenticated garbling, output-label verification, and cross-message token consistency checks.

Table 3: Classical lattice attack cost estimates (Block A) and accounted online terms $B_{\text{counted},P}$ (Block B) ($L = 1024$, $N_{\text{out}} = 64$, $Q = 2^{40}$).

Block A: Estimated classical lattice attack costs (Lattice Estimator 352ddaf4, binary secret, unlimited samples)

Lattice Instance (Applicable Modes)	d	q_{grid}	σ	Optimal Attack Family	MATZOV ($\log_2$ rop)	BDGL16 ($\log_2$ rop)
LWE-2bit (Blind / LWE)	914	2^{32}	8.94×10^{-7}	dual_hybrid	132.00	134.42
LWE-4bit (Blind / LWE)	978	2^{32}	2.96×10^{-7}	dual_hybrid	132.06	134.52
LWE-6bit (Blind / LWE)	1106	2^{32}	3.26×10^{-8}	dual_hybrid	132.30	134.42
LWE-8bit (Blind / LWE)	1267	2^{32}	2.02×10^{-9}	dual_hybrid	132.55	134.62
GLWE-2–8bit (GLWE, $k_{\text{glwe}} = 1$)	2048	2^{64}	2.85×10^{-15}	dual_hybrid	133.09	135.04

Block B: Accounted online-session terms ($\log_2 B_{\text{counted},\mathcal{S}} / \log_2 B_{\text{counted},\mathcal{C}}$; P-256 generic-group: -176.0)

PH-PBS Mode (Reference Workload)	2-bit ($\mathcal{S}/\mathcal{C}$)	4-bit ($\mathcal{S}/\mathcal{C}$)	6-bit ($\mathcal{S}/\mathcal{C}$)	8-bit ($\mathcal{S}/\mathcal{C}$)
Blind ($M_{\text{tot}} = 1024$, 1 LUT/node)	-68.04 / -68.03	-67.94 / -67.91	-67.76 / -67.68	-67.57 / -67.31
Blind ($M_{\text{tot}} = 16384$, 16/node)	-68.02 / -67.95	-67.92 / -67.63	-67.74 / -66.87	-67.55 / -65.58
LWE ($M_{\text{tot}} = 1024$ and 16384)	-68.04 / -68.04	-67.94 / -67.94	-67.76 / -67.76	-67.57 / -67.57
GLWE ($M_{\text{tot}} = 1024$ and 16384)	-66.89 / -66.89	-66.89 / -66.89	-66.89 / -66.89	-66.89 / -66.89

Note: Block A estimates classical lattice attack costs on the five native-grid instances; Block B reports selected heuristic online query-and-collision counts ($M_{\text{tot}} \in \{1{,}024, 16{,}384\}$, invariant in M_v for token modes). Formal distinguishing bounds remain (19)–(20) with p_{fail} (Section 4).

Within the static semi-honest model, three deployment boundaries define the operating contract. First, PH-PBS targets interactive workloads with an online key-holding client across two round trips per refresh, and the public execution schedule $\mathcal{G}$ (DAG topology, k, per-node table counts M_v, and deterministic inline COT refill points fixed by C_{pool}) constitutes explicit allowed leakage. Second, sessions enforce strict single-use state discipline (no reuse of one-time tokens or consumed COT tuples) and handle external transport interruptions solely via invocation-boundary termination (τ_{stop}). Third, coupling real execution to $\mathcal{F}_{\text{PH-PBS}}$ incurs the first-divergence failure probability p_{fail} bounded across all three regimes in Section 4 (Table 2) under each mode's input-domain contract ($0 \le \tilde{\mu}_v < p$ for PH-PBS-GLWE) and Assumption A1's grid-rounded Gaussian model; any statistical distance between a concrete sampler and Assumption A1 enters separately.

6 Implementation and Experimental Evaluation

We evaluate our Rust implementation of all 12 PH-PBS configurations (three modes across $k \in \{2, 4, 6, 8\}$) on a single commodity host against both a controlled same-interface interactive baseline and cross-paradigm TFHE references.

6.1 Implementation and Evaluation Setup

Implementation kernels and precomputation. The client pre-expands each binary secret bit $s_i \in \{0, 1\}$ into a full-word bitmask $\mathsf{mask}_i = 0 - s_i$ for branchless sequential dot products in LWE and GLWE encryption and token generation; during PH-PBS-LWE and PH-PBS-GLWE local evaluation, the server skips entries with $f(h) = 0$ (a branch depending solely on the server's own table values, not on client secrets μ or r). Hardware AES-NI accelerates all symmetric operations according to their key structure: 8-block batched AES-128-CTR expands small-key LWE/Blind masks and shared server garbled-input wires, 8-block batched domain-separated AES-128 encrypts PH-PBS-Blind table rows, PhasePads evaluates independent per-coordinate AES-128 keys with constant-time client accumulation, Half-Gates executes four (garbling) and two (evaluation) tweakable correlation-robust

AES-128 calls per non-free AND gate, and PH-PBS-GLWE reuses TFHE-rs's seeded random generator for polynomial mask expansion. All non-byte-aligned wire arrays (b_{sh}-bit phase and COT corrections, b_B-bit token bodies, and $(k+1)$-bit blinded table rows) are tightly bit-packed to $\lceil N_{\mathrm{elem}} b/8 \rceil$ bytes. Each party maintains a precomputed COT pool of $C_{\mathrm{pool}} = 16{,}384$ instances refilled inline via IKNP extension (262,144 B) whenever remaining balance drops below b_{sh}; in token modes, a capacity-2 background producer overlaps client token generation with network latency during sustained runs.

Host and end-to-end timing contract. All measurements run on an AMD Ryzen 7 5800H (8 cores, 16 threads, 3.20 GHz base, 16 MB L3 cache, AVX2/AES-NI) with 16 GB DDR4 RAM under 64-bit Windows 10 (Build 19045), compiled with `rustc 1.93.0` in release mode with native target-CPU optimizations. PH-PBS and GC-PrivLUT communicate over OS loopback TCP sockets (`127.0.0.1`); the channel counts exact bidirectional application bytes and brackets socket I/O with compute timers to isolate local CPU time. Single-table microbenchmarks (Table 4) average 3 independent sessions × 64 measured refreshes (after 16 warmups) plus 1 terminal decoding per session (192 refreshes and 3 terminal decodings per configuration). To account for all per-invocation preprocessing in end-to-end comparisons, we record server online time (T_{server}), client online time (T_{client}), and client token-preparation time (T_{tok}) separately and charge token preparation serially in $T_{\mathrm{local}} = T_{\mathrm{server}} + T_{\mathrm{client}} + T_{\mathrm{tok}}$ across all single-LUT latency, LAN/WAN, and RTT* metrics (PH-PBS-Blind has $T_{\mathrm{tok}} = 0$ by construction); one-time session setup and inline COT refills are accounted explicitly in Figure 2 and Table 7.

Two-tier baseline contracts. We separate same-host baselines into two tiers to distinguish architectural gains from paradigm differences. *Tier 1 (same-interface dual-privacy 2-RTT):* GC-PrivLUT shares PH-PBS-Blind's exact CtrMS frontend, PhasePads Stage 1, single-Δ_{GC} COT pool, Stage 4 seeded LWE return, and T_{local} timing contract, but moves private LUT selection inside Stage 3 via a garbled subtraction, binary multiplexer tree, and modular $\mathbb{Z}_{2p}$ blinding adder totaling $(b_{\mathrm{sh}} - 1) + k(p-1) + k$ non-free AND gates (19, 77, 399, and 2,065 gates at 2–8 bits, e.g., $17 + 2{,}040 + 8 = 2{,}065$ at 8 bits; 1 session × 32 measured refreshes after 8 warmups). *Tier 2 (cross-paradigm references):* for non-interactive 0-RTT TFHE, we run official 64-bit KSK, KS-then-PBS V1_4 parameter sets of TFHE-rs (v1.6.1) [Zam26] ($p_{\mathrm{fail}} \leq 2^{-128}$, message × carry $= 2^k$) across Classic-Gaussian [CGGI20], Classic-TUniform, and MultiBit-G3 [JP22], plus 4-bit DBM-PS16 [dRDV26] ($n = 842, N = 2048$) in the same engine and official open-source SBS+CMS [BORLT26] at 4, 6, and 8 bits ($N = 2048$, EF $\in \{2^1, 2^3, 2^5\}$). For the 8-bit TFHE-rs Classic-Gaussian and Classic-TUniform configurations ($N = 65536$), the official V1_4 parameters give raw evaluation-key sizes of 10.2 and 10.4 GiB, respectively, excluding key-generation workspace; we skipped live key generation on our 16 GB host. For client-assisted re-encryption without table privacy, we run our 4-bit reproduction of Smart and Walter [SW25] ($N = 2048$), where the client holds a 1,760.0 MiB exact KSK and spends 41.64 ms of the 41.70 ms non-terminal local time on client-side key switching (65,594 B, 1 RTT; terminal call 40.66 ms, 32,797 B, 0.5 RTT). Share-based 2PC lookup protocols (FLUTE [BHS+23], ROTL [HLL+26], FABLE [SPBZ25]) accept and return secret shares rather than server-retained native LWE ciphertexts; benchmarking them under our multi-input DAG contract would require adding an external share-to-ciphertext conversion layer, so we position them qualitatively in Section 1.

6.2 Single-LUT Refresh, Setup Cost, and Network Regimes

Keeping the garbled circuit restricted to a $(b_{\mathrm{sh}} - 1)$-AND subtraction (11, 13, 15, and 17 gates at 2–8 bits) holds PH-PBS-Blind's combined local computation to 0.0426–0.0595 ms

Table 4: Single-LUT microbenchmarks and 1.5-RTT terminal decoding across 12 PH-PBS sets and Tier-1 GC-PrivLUT (Block A), with 4/8-bit micro-breakdowns and output phase variances (Block B).

Block A: Single-LUT microbenchmarks and 1.5-RTT terminal decoding (local elapsed CPU time excl. socket I/O)

Mode / Baseline	k (bits)	State ↓ ($\mathcal{S}$/$\mathcal{C}$, MiB)	Srv. ↓ (ms)	Cli. ↓ (ms)	Tok. ↓ (ms)	Total T_{local} ↓ (mean ± std, ms)	Online Comm. ↓ ($\mathcal{S}\to\mathcal{C}$/$\mathcal{C}\to\mathcal{S}$, B)	Term. Dec. ↓ (ms / B)
PH-PBS-Blind	2	1.57/1.08	0.0266	0.0160	0.0000	**0.0426 ± 0.0143**	1,745/22 (**1,767**)	0.0340/1,747
PH-PBS-Blind	4	1.66/1.12	0.0312	0.0193	0.0000	**0.0506 ± 0.0140**	2,158/22 (**2,180**)	0.0332/2,152
PH-PBS-Blind	6	1.83/1.21	0.0305	0.0163	0.0000	**0.0468 ± 0.0138**	2,768/22 (**2,790**)	0.0379/2,716
PH-PBS-Blind	8	2.05/1.32	0.0364	0.0232	0.0000	**0.0595 ± 0.0116**	3,703/23 (**3,726**)	0.0451/3,420
PH-PBS-LWE	2	1.58/1.09	0.0282	0.0119	0.0067	0.0468 ± 0.0106	1,743/30 (1,773)	0.0318/1,747
PH-PBS-LWE	4	1.72/1.18	0.0418	0.0133	0.0222	0.0774 ± 0.0098	2,148/68 (2,216)	0.0335/2,152
PH-PBS-LWE	6	2.10/1.48	0.1044	0.0118	0.0941	0.2103 ± 0.0203	2,712/244 (2,956)	0.0308/2,716
PH-PBS-LWE	8	3.28/2.56	0.4027	0.0190	0.4137	0.8353 ± 0.0528	3,415/1,045 (4,460)	0.0598/3,420
PH-PBS-GLWE	2	3.15/1.91	0.0991	0.0232	0.0612	0.1834 ± 0.0225	4,020/32 (4,052)	0.0657/4,024
PH-PBS-GLWE	4	3.15/1.91	0.1035	0.0186	0.0713	0.1934 ± 0.0344	4,596/70 (4,666)	0.0475/4,600
PH-PBS-GLWE	6	3.15/1.91	0.1285	0.0249	0.0937	0.2471 ± 0.0507	5,172/247 (5,419)	0.0668/5,177
PH-PBS-GLWE	8	3.15/1.91	0.1970	0.0257	0.1828	0.4055 ± 0.0573	5,748/1,303 (7,051)	0.0754/5,753
GC-PrivLUT	2	1.57/1.08	0.0237	0.0181	0.0000	0.0419 ± 0.0030	2,001/22 (2,023)	–
GC-PrivLUT	4	1.66/1.12	0.0430	0.0293	0.0000	0.0723 ± 0.0114	4,198/22 (4,220)	–
GC-PrivLUT	6	1.83/1.21	0.0983	0.0747	0.0000	0.1730 ± 0.0218	15,002/22 (15,024)	–
GC-PrivLUT	8	2.05/1.32	0.3768	0.3269	0.0000	0.7038 ± 0.0300	68,953/23 (68,976)	–

Block B: Per-step micro-breakdown (μs) and empirical vs. analytical output phase variance (4-bit and 8-bit)

Mode (k)	$\mathcal{S}$-CtrMS	$\mathcal{S}$-Phase	$\mathcal{S}$-GC	$\mathcal{S}$-Eval	$\mathcal{C}$-Phase	$\mathcal{C}$-GC	$\mathcal{C}$-Enc	Emp. $\widehat{\mathrm{Var}}(e_{\text{out}})$	Ana. σ^2_{out}	Ratio
Blind (4-bit)	3.27	17.56	6.57	2.64	10.14	4.44	3.18	8.50×10^{-14}	8.78×10^{-14}	0.97×
LWE (4-bit)	2.27	15.38	3.18	19.72	9.16	2.92	0.00	9.47×10^{-11}	1.09×10^{-10}	0.87×
GLWE (4-bit)	4.56	24.46	3.54	69.42	14.06	3.42	0.00	4.09×10^{-13}	3.67×10^{-13}	1.11×
Blind (8-bit)	2.41	19.40	9.73	3.65	12.87	5.77	2.93	5.99×10^{-18}	4.10×10^{-18}	1.46×
LWE (8-bit)	2.77	20.57	3.95	373.80	14.29	3.56	0.00	2.10×10^{-11}	2.28×10^{-11}	0.92×
GLWE (8-bit)	4.74	29.74	4.17	156.84	20.66	3.94	0.00	3.94×10^{-19}	3.83×10^{-19}	1.03×

Note: PH-PBS averages 3×64 refreshes and 3 terminal decodings; GC-PrivLUT averages 1×32 refreshes. One-time setup takes 71.5–78.1 ms ($\mathcal{S}$) and 41.1–44.9 ms ($\mathcal{C}$), transferring 326.0–341.1 KiB (Blind/LWE/GC-PrivLUT) or 377.5 KiB (GLWE).

across all four bit-widths (Table 4, Block A). At 8-bit precision, moving private table selection outside the circuit cuts combined local compute from 0.7038 ms to 0.0595 ms (11.82× speedup) and online traffic from 68,976 B to 3,726 B (a 94.6% reduction, with only 23 B uploaded by the client) relative to same-interface GC-PrivLUT. For designated final outputs, 1.5-RTT terminal decoding omits Stage 4 re-encryption and blinded tables, completing in 0.0308–0.0754 ms and 1,747–5,753 B across all 12 configurations. The per-step breakdown (Table 4, Block B) explains the precision crossover between the two token modes: CtrMS and subtraction garbling/evaluation each remain below 10 μs, while at 8 bits ($p = 256$), PH-PBS-LWE expands and multiplies 256 masks of dimension $n = 1267$ ($T_{\text{local}} = 0.8353$ ms), whereas PH-PBS-GLWE expands one degree-2048 ring mask and evaluates 256 projected active bodies, cutting T_{local} in half to 0.4055 ms. Across the 4-bit and 8-bit rows in Block B, empirical output phase variances stay within 0.87×–1.11× of the Subsection 4.2 analytical variance model except for a 1.46× draw on the 192-sample 8-bit PH-PBS-Blind run (5.99×10^{-18} vs. 4.10×10^{-18}). In our implementation, Stage 4 unblinding and 2^{32} lifting satisfy the bit-exact error identity $\mathrm{ctr}_q(b_{\text{out}} - \langle \mathbf{a}_{\text{out}}, \mathbf{s} \rangle - \Delta f(\mu)) = 2^{32}\mathrm{ctr}_{2^{32}}(e^{(32)})$ by construction; an independent 192-sample rerun of the identical binary over the full 8-bit Blind refresh path yields 3.70×10^{-18} (0.90×, with all 12 configurations in 0.84×–1.18×), and ten consecutive runs of our built-in 4,096-sample fresh-encryption regression test (40,960 samples per precision) all stay within $[0.85, 1.15]$. These follow-up checks show no persistent variance inflation and are consistent with a finite-sample tail draw.

Table 5: Tiered single-LUT comparison across 2–8 bits: Tier-1 same-interface vs. Tier-2 cross-paradigm references, LAN/WAN latency, and break-even RTT*.

k	Scheme / Baseline	Tier	Key/State ↓ (MiB)	1T Local ↓ (ms)	Comm. ↓ (B)	LAN ↓ (ms)	WAN ↓ (ms)	RTT* ↑ (100M / 1G, ms)
2	PH-PBS-Blind	T1	**1.57/1.08**	**0.0426**	**1,767**	**1.06**	100.18	–
	GC-PrivLUT	T1	1.57/1.08	0.0419	2,023	1.06	100.20	–
	TFHE-rs Classic-Gaussian	T2-NI	121.0	12.13	0	12.13	12.13	5.97/**6.04**
	TFHE-rs Classic-TUniform	T2-NI	127.1	12.58	0	12.58	12.58	6.20/**6.26**
	TFHE-rs MultiBit-G3	T2-NI	182.2	18.74	0	18.74	18.74	9.28/**9.34**
4	PH-PBS-Blind	T1	**1.66/1.12**	**0.0506**	**2,180**	**1.07**	100.22	–
	GC-PrivLUT	T1	1.66/1.12	0.0723	4,220	1.11	100.41	–
	DBM-PS16 [dRDV26]	T2-NI	118.5	15.64	0	15.64	15.64	7.71/**7.78**
	TFHE-rs Classic-Gaussian	T2-NI	121.9	16.26	0	16.26	16.26	8.02/**8.09**
	TFHE-rs Classic-TUniform	T2-NI	114.8	16.82	0	16.82	16.82	8.30/**8.38**
	SBS+CMS [BORLT26]	T2-NI	120.9	20.61	0	20.61	20.61	10.19/**10.27**
	TFHE-rs MultiBit-G3	T2-NI	380.6	34.89	0	34.89	34.89	17.33/**17.41**
	Smart–Walter [SW25] (1 RTT)	T2-CE	1,760.0	41.70	65,594	42.73	96.95	–
	Smart–Walter [SW25] (0.5 RTT)	T2-CE	1,760.0	40.66	32,797	41.17	68.28	–
6	PH-PBS-Blind	T1	**1.83/1.21**	**0.0468**	**2,790**	**1.07**	**100.27**	–
	GC-PrivLUT	T1	1.83/1.21	0.1730	15,024	1.29	101.37	–
	SBS+CMS [BORLT26]	T2-NI	245.0	92.71	0	92.71	92.71	46.22/**46.32**
	TFHE-rs Classic-Gaussian	T2-NI	943.6	123.45	0	123.45	123.45	61.59/**61.69**
	TFHE-rs Classic-TUniform	T2-NI	875.4	127.98	0	127.98	127.98	63.85/**63.95**
	TFHE-rs MultiBit-G3	T2-NI	3,332.0	291.80	0	291.80	291.80	145.77/**145.87**
8	PH-PBS-Blind	T1	**2.05/1.32**	**0.0595**	**3,726**	**1.09**	**100.36**	–
	GC-PrivLUT	T1	2.05/1.32	0.7038	68,976	2.26	106.22	–
	SBS+CMS [BORLT26]	T2-NI	435.2	390.02	0	390.02	390.02	194.83/**194.97**
	TFHE-rs Classic-Gaussian	T2-NI	10,434.5[a]	–[a]	0	–	–	–
	TFHE-rs Classic-TUniform	T2-NI	10,615.0[a]	–[a]	0	–	–	–

Note: T1 = same-interface dual-privacy 2-RTT; T2-NI = non-interactive 0-RTT TFHE; T2-CE = client plaintext evaluation (no table privacy). Key/State = BSK + KSK (T2-NI), client KSK (T2-CE), or $\mathcal{S}/\mathcal{C}$ state (T1). LAN: 0.5 ms/1 Gbps; WAN: 50 ms/100 Mbps ($R_{\text{rtt}} = 2$ for T1; 1 and 0.5 for Smart–Walter). [a]Analytical V1_4 key sizes (10.2 and 10.4 GiB, $N = 65536$, excl. keygen workspace); live keygen skipped on our 16 GB host.

Across 2–8 bits, PH-PBS replaces hundred-MiB to multi-GiB evaluation keys with 1.08–3.28 MiB of persistent session state per party and delivers broad network operating envelopes (Table 5). Under the steady-state model $T_{\text{e2e}}(\text{RTT}, B_{\text{net}}) = T_{\text{local}} + 8B/B_{\text{net}} + R_{\text{rtt}}\,\text{RTT}$ ($R_{\text{rtt}} = 2$ for PH-PBS), PH-PBS-Blind achieves 1.057–1.089 ms on a LAN (0.5 ms RTT, 1 Gbps), compared with 12.13–390.02 ms for single-threaded non-interactive TFHE baselines and 42.73 ms for 4-bit Smart–Walter (which also requires a 1,760.0 MiB client KSK and exposes intermediate plaintexts and tables). The steady-state break-even threshold RTT* at 1 Gbps spans 6.04–9.34 ms at 2 bits and 7.78–17.41 ms at 4 bits (covering datacenter, LAN, and edge links), then widens sharply to 46.32–145.87 ms at 6 bits and 194.97 ms against 8-bit SBS+CMS (194.78 ms for PH-PBS-GLWE and 194.57 ms for PH-PBS-LWE). Accordingly, even over a 50 ms, 100 Mbps WAN link, 8-bit PH-PBS-Blind completes in 100.36 ms (3.89× faster than single-threaded 8-bit SBS+CMS at 390.02 ms) and 6-bit PH-PBS-Blind (100.27 ms) still outperforms 6-bit TFHE-rs Classic-Gaussian (123.45 ms), whereas at 2–4 bits the 2-RTT WAN floor (≈ 100.2 ms) exceeds all measured non-interactive PBS baselines (12.1–34.9 ms), which marks a clear precision-dependent deployment boundary.

Accounting for one-time session setup and periodic COT pool refills reinforces this advantage over long horizons (Figure 2). Including base OTs, PhasePads OT extension, initial 16,384-COT pool population, and 1 Gbps transfer, 8-bit PH-PBS-Blind and PH-PBS-GLWE complete session setup in 0.12 s (0.34 and 0.38 MiB), 11×–388× faster than measured TFHE-rs key generation plus 1 Gbps upload at 4–6 bits (1.37–47.1 s for 118.5 MiB–3.3 GiB) and over 720× below the 87.5 s 1 Gbps upload lower bound alone for the 10.2 GiB 8-bit TFHE-rs key (Figure 2a). Over sustained 8-bit execution including setup, online messages, and exact periodic 262,144 B COT refills (Figure 2b), PH-PBS-Blind transfers only 4.2 MiB over $L = 1{,}024$ refreshes (including setup and one COT refill)

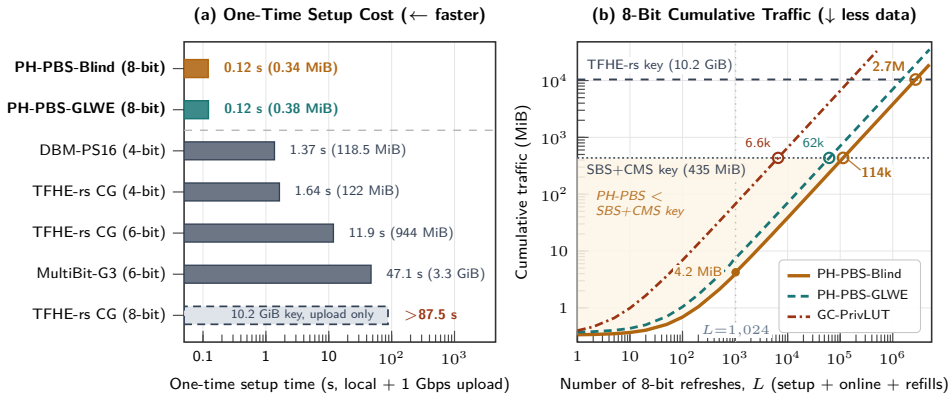

Figure 2: One-time setup cost and 8-bit cumulative traffic amortization. (a) Setup time (local key/OT generation plus 1 Gbps upload; dashed 8-bit TFHE-rs bar shows 1 Gbps key upload alone). (b) Cumulative 8-bit traffic (setup, online messages, and periodic 262,144 B COT refills) vs. upfront evaluation-key uploads.

and remains below a single 8-bit SBS+CMS key upload (435.2 MiB) for 114k consecutive refreshes (17.3× longer than GC-PrivLUT's 6.6k horizon, with PH-PBS-GLWE reaching 62k), and below a single 8-bit TFHE-rs key (10.2 GiB) for 2.7M refreshes.

6.3 Multi-LUT Scaling and Projected GLWE Validation

When a refresh node evaluates $M_v \in \{1, 2, 4, 8, 16\}$ parallel private tables on the same input ciphertext (1 session × 24 measured refreshes after 4 warmups), PH-PBS-LWE and PH-PBS-GLWE hold online communication strictly constant and confine marginal work to server-side linear accumulation (Table 6). Because both token modes transmit one PhasePads message, one single-use shift token, one subtraction circuit, and k_{out} output selection bits per node regardless of M_v, online traffic remains invariant across M_v, amortizing down to 278.8 and 440.7 B/LUT at 8 bits ($M_v = 16$). By contrast, PH-PBS-Blind shares the subtraction circuit across tables but transmits M_v blinded tables in Stage 3 and M_v seeded ciphertexts in Stage 4 (growing linearly by +30 B and +308 B per extra table at 4 and 8 bits); at 8 bits, PH-PBS-LWE crosses below PH-PBS-Blind at $M_v = 4$ (4,460 vs. 4,650 B) and cuts traffic by 46.6% at $M_v = 16$ (4,460 vs. 8,346 B). Thus PH-PBS-Blind minimizes compute and traffic on single-table (and low-fan-out compute-bound) nodes, PH-PBS-LWE minimizes traffic on full-domain multi-table nodes, and PH-PBS-GLWE targets 8-bit half-domain nodes requiring tighter fresh output noise for wider affine budgets.

Controlled checks also verify the two projected GLWE mechanisms of Subsection 3.4. At each precision $k \in \{2, 4, 6, 8\}$, ProjEval matches the dense reference $\text{Ext}_0(Q_f X^{\alpha} \hat{C})$ across all $N+1 = 2{,}049$ extracted words in 64/64 trials, including when the $N-p$ discarded body coordinates hold arbitrary nonzero values (Proposition 1). Across 128 timed trials per precision (after 8 warmups), sampling noise solely on the retained active set I_ρ (Lemma 1) accelerates client token generation by 2.10–4.79× relative to full-polynomial noise sampling (0.0588–0.1732 ms vs. 0.2815–0.3645 ms).

6.4 Multi-Input DAGs, Sustained Refill, and Multi-Core Throughput

End-to-end circuit and long-session workloads verify homomorphic composition after refresh and quantify COT pool replenishment (Table 7, Block A). On a 20-layer, width-8,

Table 6: Same-input multi-LUT scaling across $M_v \in \{1, 2, 4, 8, 16\}$ parallel private tables at 4 and 8 bits. Token modes keep online traffic invariant in M_v; PH-PBS-Blind scales linearly.

k (bits)	Mode	M_v=1 ↓ (ms)	M_v=2 ↓ (ms)	M_v=4 ↓ (ms)	M_v=8 ↓ (ms)	M_v=16 ↓ (ms)	$\mathcal{S}/\mathcal{C}$/Tok ↓ (M_v=16, ms)	Online Comm. ↓ (B/node)
4	Blind	0.045	0.048	0.075	0.094	0.137	0.10/0.04/0.00	$2{,}180 + 30(M_v{-}1)$
	LWE	0.075	0.084	0.088	0.105	0.160	0.12/0.02/0.02	**2,216**
	GLWE	0.239	0.190	0.249	0.308	0.381	0.30/0.02/0.07	**4,666**
8	Blind	0.056	0.063	0.097	0.141	0.253	0.18/0.07/0.00	$3{,}726 + 308(M_v{-}1)$
	LWE	0.852	0.809	0.887	1.003	1.285	0.83/0.02/0.43	**4,460**
	GLWE	0.424	0.511	0.838	1.173	1.954	1.70/0.03/0.22	**7,051**

Table 7: 20-layer multi-input DAG and 1,400-layer COT refill run (Block A), and concurrent throughput across 1–8 workers with link ceilings (Block B).

Block A: 20-layer width-8 fan-in-4 DAG (160 refreshes + 8 terminal outputs) and 1,400-layer sustained run

Mode (k)	20-Layer Split ↓ ($\mathcal{S}/\mathcal{C}$/Tok, ms)	Per Refresh ↓ (ms / B)	Refills (in./tot.)	Wall Time ↓ (ms/layer)	Steady ↓ (B/layer)	Amort. Total ↓ (B/layer, Δ)	Check
Blind (4-bit)	4.15/3.05/0.00	0.0450/2,287.6	1/2	0.1393	2,180	**2,368.8** (+188.8)	PASS
LWE (4-bit)	6.40/2.42/3.85	0.0792/2,323.6	1/2	0.1569	2,216	**2,404.8** (+188.8)	PASS
GLWE (4-bit)	21.19/3.71/15.06	0.2497/4,896.0	1/2	0.2237	4,666	**4,856.5** (+190.5)	PASS
Blind (8-bit)	6.65/4.79/0.00	0.0715/3,897.0	1/2	0.1585	3,726	**3,915.7** (+189.7)	PASS
LWE (8-bit)	65.70/3.35/65.09	0.8384/4,631.0	1/2	0.5268	4,460	**4,649.7** (+189.7)	PASS
GLWE (8-bit)	30.06/3.93/28.75	0.3922/7,338.7	1/2	0.3259	7,051	**7,242.4** (+191.4)	PASS

Block B: Concurrent throughput across $W \in \{1, 2, 4, 8\}$ workers (loopback ops/s ↑; token modes report online / E2E incl. live TokPrep)

k	Scheme / Baseline	$W = 1$ ↑	$W = 2$ ↑	$W = 4$ ↑	$W = 8$ ↑	1G / 100M Cap
2	PH-PBS-Blind	**6,860**	**12,784**	**22,304**	**38,493**	70.7k/7.07k
	PH-PBS-LWE (onl./E2E)	6,687/6,725	14,344/12,768	25,076/24,198	**39,836/34,518**	70.5k/7.05k
	PH-PBS-GLWE (onl./E2E)	4,010/4,417	7,887/7,521	15,428/14,996	22,970/17,921	30.8k/3.08k
	TFHE-rs Classic-Gaussian	82.4	158.5	306.6	425.4	–
	TFHE-rs Classic-TUniform	79.5	151.1	284.6	383.1	–
	TFHE-rs MultiBit-G3	53.4	97.2	166.1	268.6	–
4	PH-PBS-Blind	**6,399**	**11,970**	**24,030**	**37,344**	57.3k/5.73k
	PH-PBS-LWE (onl./E2E)	6,324/5,910	11,270/11,112	23,581/19,904	**38,356/29,611**	56.4k/5.64k
	PH-PBS-GLWE (onl./E2E)	4,435/4,444	8,603/8,424	15,440/14,788	21,718/18,046	26.8k/2.68k
	DBM-PS16 [dRDV26]	64.0	122.2	199.6	266.1	–
	TFHE-rs Classic-Gaussian	61.5	117.1	225.6	315.6	–
	TFHE-rs Classic-TUniform	59.4	112.4	216.1	285.9	–
	TFHE-rs MultiBit-G3	28.7	54.2	92.7	143.7	–
6	PH-PBS-Blind	**6,384**	**12,889**	**23,999**	**37,020**	44.8k/4.48k
	PH-PBS-LWE (onl./E2E)	4,466/4,577	8,758/8,351	16,729/13,600	25,859/18,778	42.3k/4.23k
	PH-PBS-GLWE (onl./E2E)	3,988/3,994	7,846/7,010	13,972/13,913	20,331/15,685	23.1k/2.31k
	TFHE-rs Classic-Gaussian	8.1	15.3	25.6	35.1	–
	TFHE-rs Classic-TUniform	7.8	14.9	23.7	21.0	–
	TFHE-rs MultiBit-G3	3.4	5.7	8.0	4.9	–
8	PH-PBS-Blind	**5,578**	**11,263**	**21,059**	**34,242**	33.5k/3.35k
	PH-PBS-LWE (onl./E2E)	1,846/1,790	3,457/3,466	6,665/5,496	9,370/4,138	28.0k/2.80k
	PH-PBS-GLWE (onl./E2E)	3,078/2,839	6,077/5,476	10,082/9,919	**14,663/11,741**	17.7k/1.77k

Note: Block A "Per Refresh" divides 20-layer totals (incl. 8 terminal decodings) by 160 refreshes; "Refills (1/2)" = 1 inline COT refill (262,144 B) + setup; "Amort. Total" averages 1,400 refreshes + 1 refill + 1 terminal decoding. Block B W = independent pairs (64 refreshes/pair, PH-PBS) or Rayon threads (TFHE-rs/DBM); "1G / 100M Cap" = $B_{\text{net}}/(8B)$.

fan-in-4 multi-input affine–LUT DAG (160 refreshes and 8 terminal decodings; unit weights $w_{v,h} = 1$, random biases $\beta_v \in \{0, 1, 2, 3\}$, inputs and LUT outputs in $[0, p/4)$), all six 4-bit and 8-bit configurations match the plaintext reference on 8/8 terminal outputs, and both per-refresh local times (0.0450–0.8384 ms) and per-refresh byte counts (2,287.6–7,338.7 B) (each dividing total circuit cost including the 8 terminal decodings by 160 refreshes) closely track Table 4. Across a 1,400-layer sequential refresh run consuming

$1{,}400\, b_{\mathrm{sh}} \in [19{,}600, 28{,}000]$ COT instances, each configuration performs one initial setup fill, triggers one inline IKNP COT refill of 262,144 B (187.25 B/layer), and concludes with one terminal decoding (1.54–4.11 B/layer), raising amortized session traffic by only +188.8 to +191.4 B/layer over steady-state payloads; with the capacity-2 streaming token queue active, 8-bit wall-clock time per layer (including loopback transport and the inline refill) stays at 0.5268 ms for PH-PBS-LWE and 0.3259 ms for PH-PBS-GLWE.

Across $W \in \{1, 2, 4, 8\}$ concurrent workers (Table 7, Block B), loopback PH-PBS-Blind scales to 34.2k–38.5k ops/s across 2–8 bits, while 8-bit PH-PBS-GLWE reaches 14,663 (online) and 11,741 ops/s (end-to-end with live token generation), compared with 315.6 and 35.1 ops/s for 8-thread 4-bit and 6-bit TFHE-rs Classic-Gaussian. Because loopback sockets impose no physical link cap, sustainable throughput over a real shared link is bounded by the minimum of CPU capacity and the link ceiling $B_{\mathrm{net}}/(8B)$ in the last column (for instance, 8-bit PH-PBS-Blind saturates a 1 Gbps link at 33.5k ops/s and a 100 Mbps link at 3.35k ops/s). Two non-monotonic drops from $W = 4$ to $W = 8$ are consistent with CPU core oversubscription and cache/memory-bandwidth pressure on our 8-core host: in 8-bit PH-PBS-LWE end-to-end mode (5,496 → 4,138 ops/s), each session pair executes concurrent server evaluation and live token generation (16 compute-heavy threads on 8 physical cores), whereas in 6-bit TFHE-rs MultiBit-G3 (8.0 → 4.9 ops/s) and Classic-TUniform (23.7 → 21.0 ops/s), concurrent reads span 875 MiB–3.33 GiB evaluation keys against a 16 MB L3 cache.

7 Conclusion

We presented PH-PBS, a two-round phase-hiding programmable bootstrapping suite that preserves native server-side LWE ciphertexts for interactive multi-input affine–LUT DAGs while eliminating blind rotation, key switching, and their evaluation keys. By moving linear phase sharing and private table backends outside the garbled circuit and confining garbling to logarithmic subtraction, PH-PBS supports single-table refresh (PH-PBS-Blind), constant-communication multi-table fan-out (PH-PBS-LWE and projected PH-PBS-GLWE), and 1.5-RTT terminal decoding without exposing intermediate plaintexts or private tables. We proved full-session static semi-honest simulation security (Section 5) and certified task failure bounds below 2^{-316} over random keys and execution for fixed non-adaptive tasks (1,024 refreshes, 64 terminal decodings) under our grid-rounded Gaussian model and certified weight budgets, alongside ideal-mask and fixed-key bounds in Section 4. Across 2–8 bits, our Rust implementation holds persistent session state to 1.08–3.28 MiB (0.12 s setup), delivers an 11.82× local speedup and 94.6% traffic reduction over 8-bit GC-PrivLUT, and achieves a 194.97 ms 1 Gbps break-even RTT against single-threaded 8-bit SBS+CMS.

References

[AGHV25] Adi Akavia, Craig Gentry, Shai Halevi, and Margarita Vald. Achievable CCA2 relaxation for homomorphic encryption. *Journal of Cryptology*, 38(1):5, 2025. doi:10.1007/s00145-024-09526-1.

[ALSZ17] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions. *Journal of Cryptology*, 30(3):805–858, 2017. doi:10.1007/s00145-016-9236-6.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. doi:10.1515/jmc-2015-0016.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 10–24. SIAM, 2016. doi:10.1137/1.9781611974331.ch2.

[BHS+23] Andreas Brüggemann, Robin Hundt, Thomas Schneider, Ajith Suresh, and Hossein Yalame. FLUTE: Fast and secure lookup table evaluations. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 515–533. IEEE, 2023. doi:10.1109/SP46215.2023.10179345.

[BORLT26] Loris Bergerat, Jean-Baptiste Orfila, Adeline Roux-Langlois, and Samuel Tap. Accelerating TFHE with sorted bootstrapping techniques. In *Advances in Cryptology – ASIACRYPT 2025*, volume 16251 of *Lecture Notes in Computer Science*, pages 68–100. Springer, 2026. doi:10.1007/978-981-95-5122-4_3.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020. doi:10.1007/s00145-019-09319-x.

[CIM19] Sergiu Carpov, Malika Izabachène, and Victor Mollimard. New techniques for multi-value input homomorphic evaluation and applications. In *Topics in Cryptology – CT-RSA 2019*, volume 11405 of *Lecture Notes in Computer Science*, pages 106–126. Springer, 2019. doi:10.1007/978-3-030-12612-4_6.

[CLOT21] Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for TFHE. In *Advances in Cryptology – ASIACRYPT 2021*, volume 13092 of *Lecture Notes in Computer Science*, pages 670–699. Springer, 2021. doi:10.1007/978-3-030-92078-4_23.

[CO15] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In *Progress in Cryptology – LATINCRYPT 2015*, volume 9230 of *Lecture Notes in Computer Science*, pages 40–58. Springer, 2015. doi:10.1007/978-3-319-22174-8_3.

[dRDV26] Thomas de Ruijter, Jan-Pieter D'Anvers, and Ingrid Verbauwhede. Don't be mean: Reducing approximation noise in TFHE through mean compensation. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2026(1):82–104, 2026. doi:10.46586/tches.v2026.i1.82-104.

[DS16] Léo Ducas and Damien Stehlé. Sanitization of FHE ciphertexts. In *Advances in Cryptology – EUROCRYPT 2016*, volume 9665 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2016. doi:10.1007/978-3-662-49890-3_12.

[GBA21] Antonio Guimarães, Edson Borin, and Diego F. Aranha. Revisiting the functional bootstrap in TFHE. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2021(2):229–253, 2021. doi:10.46586/tches.v2021.i2.229-253.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 169–178. ACM, 2009. doi:10.1145/1536414.1536440.

[GKW+20] Chun Guo, Jonathan Katz, Xiao Wang, Chenkai Weng, and Yu Yu. Better concrete security for Half-Gates garbling (in the multi-instance setting). In *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 793–822. Springer, 2020. doi:10.1007/978-3-030-56880-1_28.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.

[HjLHD22] Zhicong Huang, Wen jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 809–826. USENIX Association, 2022. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/huang-zhicong.

[HLL+26] Xiaoyang Hou, Jian Liu, Jingyu Li, Jiawen Zhang, Kui Ren, and Chun Chen. ROTL: Faster lookup table evaluation. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2026(2):499–516, 2026. doi:10.56553/popets-2026-0058.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer, 2003. doi:10.1007/978-3-540-45146-4_9.

[JP22] Marc Joye and Pascal Paillier. Blind rotation in fully homomorphic encryption with extended keys. In *Cyber Security, Cryptography, and Machine Learning (CSCML 2022)*, volume 13301 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2022. doi:10.1007/978-3-031-07689-3_1.

[JVC18] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669. USENIX Association, 2018. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/juvekar.

[KS08] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In *Automata, Languages and Programming (ICALP 2008)*, volume 5126 of *Lecture Notes in Computer Science*, pages 486–498. Springer, 2008. doi:10.1007/978-3-540-70583-3_40.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010. doi:10.1007/978-3-642-13190-5_1.

[MAT22] MATZOV. Report on the security of LWE: Improved dual lattice attack. Zenodo Technical Report, 2022. doi:10.5281/zenodo.6412487.

[MHW+24] Shihe Ma, Tairong Huang, Anyu Wang, Qixian Zhou, and Xiaoyun Wang. Fast and accurate: Efficient full-domain functional bootstrap and digit decomposition for homomorphic computation. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2024(1):592–616, 2024. doi:10.46586/tches.v2024.i1.592-616.

[MLS+20] Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. Delphi: A cryptographic inference service for neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2505–2522. USENIX Association, 2020. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/mishra.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009. doi:10.1145/1568318.1568324.

[SPBZ25] Zhengyuan Su, Qi Pang, Simon Beyzerov, and Wenting Zheng. FABLE: Batched evaluation on confidential lookup tables in 2PC. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 1945–1964. USENIX Association, 2025. URL: https://www.usenix.org/conference/usenixsecurity25/presentation/su-zhengyuan.

[SW25] Nigel P. Smart and Michael Walter. Error-simulatable sanitization for TFHE and applications. *IACR Communications in Cryptology*, 2(3):1, 2025. doi:10.62056/a0lmpgxq.

[Zam26] Zama. TFHE-rs, version 1.6.1. https://github.com/zama-ai/tfhe-rs/releases/tag/tfhe-rs-1.6.1, 2026.

[ZRE15] Samee Zahur, Mike Rosulek, and David Evans. Two halves make a whole: Reducing data transfer in garbled circuits using half gates. In *Advances in Cryptology – EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 220–250. Springer, 2015. doi:10.1007/978-3-662-46803-6_8.