Where the Information Lives: A Key-Free Observability Decomposition for CKKS Hardware Monitors

Bader Alissaei

VaultBytes Innovations Ltd
b@vaultbytes.com

Abstract. Hardware accelerators for the CKKS fully homomorphic encryption scheme increasingly embed on-die monitors: integrity checkers for faulty datapath stages and budget gauges for deciding when precision is being exhausted. Such monitors inspect ciphertext state without the secret key. We ask what such a key-free *coefficient observer* can learn under an explicit public-metadata model and an *exogenous additive-deviation* fault model (deviations independent of the key, plaintext, and errors), and give a three-region decomposition of the information available to it. *First, ciphertext content is masked.* Under multi-sample circular decision-RLWE, no efficient key-free observer of the ciphertext coefficients has non-negligible advantage in distinguishing hidden plaintext content, or predicates of the residual within-level RLWE noise whose conditioned ensembles remain within the assumed RLWE-pseudorandom class. *Second, the coarse budget signal is public.* For a fixed public parameter set and public evaluation schedule, the key-free part of the remaining precision estimate is determined by metadata such as level, modulus chain, scale, and operation history; the realized fine residual remains masked. *Third, public relations are verifiable.* The fixed public linear maps of the CKKS datapath can be checked key-free by a precomputed random projection; for deviations nonzero modulo an independently challenged prime field of size κ, the one-sided algebraic false-accept probability is at most $1/\kappa$.
We then prove a *completeness theorem*—an upper bound on a key-free observer's view, not a new hardness result. Relative to the checked relation family and an exogenous additive-deviation model, every efficient key-free observer has a simulator that reproduces its output distribution from only the public metadata and the local computation-deviation transcript. The transcript given to the simulator contains the full local deviation δ; a real relation check may surface only a projection of it, so the bound deliberately over-approximates what actually leaks. Thus, up to negligible terms, a key-free monitor's useful view is simulable from the metadata plus the local deviation transcript; the content-bearing part of the ciphertext coefficients contributes no additional information *about hidden plaintexts, secrets, or admissible within-level noise predicates* (it may still contribute content-free artifacts such as the modulo bias of Remark 1). The fault-free case is the usual simulation consequence of semantic security; the refinement is the faulty case: the extra information captured by the ideal transcript is the deviation itself, not plaintext content or fine noise.

This decomposition gives a design rule for CKKS hardware monitors: read budget information from metadata and spend integrity effort on public relation checks, rather than on coefficient-magnitude noise gauges. The security rests on the reduction, not on the experiments; we provide reproducible sanity checks consistent with the model, with experiments reporting an explicit null-advantage baseline. Hand-built and learned key-free observers remain at the estimator's bias floor for within-level noise across $N \in \{256, 512, 1024\}$, and for plaintext content except for one small-sample cell; the same statistic after partial decryption reaches advantage 0.998 (at N=512). Public metadata separates metadata-defined coarse budget classes at advantage 1.0, while a precomputed relation check attains false-accept probability $1/\kappa$ for faults not aligned to the tested prime (unconditional injection measured $\approx 2/\kappa$; §9).

1 Introduction

As CKKS [1] moves onto fixed-function silicon, designers add on-accelerator monitors that watch ciphertext state as it streams through the datapath: *integrity* monitors for transient or injected faults and *noise-budget* gauges that decide when to bootstrap. The defining constraint is that the accelerator processes ciphertexts without the secret key—hence without the plaintext or the true noise. The practical question is then:

> *What can an on-die monitor learn about a CKKS ciphertext from the data it sees, without the secret key—and where does the information a monitor needs actually live?*

The naive temptations are well known and partly wrong: build a cheap noise gauge by reading a ciphertext limb's magnitude (it cannot work), or assume an integrity check must cost a recomputation (it need not). We ask what a key-free monitor can learn *under an explicit coefficient-observer and exogenous-fault model*, give a decomposition that settles both temptations, says where the needed information *is*, and—via a completeness theorem—bounds what any key-free monitor can extract under that model. In one line: a key-free CKKS monitor can usefully read only two things—the public metadata (for the noise budget) and the deviations exposed by checking public relations (for integrity); the ciphertext's content bytes tell it nothing it can act on.

This is not a new hardness result. We state this up front to avoid any misreading: Theorem 1 (region i) is the monitor-facing restatement of CKKS semantic security under RLWE—nothing about the hardness of CKKS is claimed or improved. The paper's value is what is built *on top* of that fact: the three-region decomposition, the public-metadata budget channel, and the deviation-transcript completeness boundary.

Scope and honest positioning. The masking fact (region i) follows from RLWE and we present it as the monitor-facing consequence. The relation-checking

fact (region iii) is classical Freivalds/ABFT [3,4], developed for lattice/NTT hardware [5,6]; we attribute it and use it only to bound the design space. *Our contributions are:* (a) region (ii), the coarse/fine budget decomposition—the needed information is public metadata, the masked part is the within-level noise; (b) the *completeness theorem* (Section 8), which turns the three-region picture from a list into a proved boundary via a simulation argument; and (c) an experimentally calibrated demonstration, including a *learned* observer and a multi-N sweep, with a null baseline. The aim is a clean, correct, useful structural statement and a design rule, not a new mechanism.

2 Preliminaries

Let $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ for power-of-two N. In practice $q = \prod_i q_i$ is an RNS product of word primes and the current *level* is the number of remaining q_i; κ denotes a single prime (sub-)modulus used by the relation check. A CKKS ciphertext encrypting message m under secret s is

$$\mathrm{ct} = (b, a) \in R_q^2, \qquad b = -a \cdot s + e + \Delta \cdot m \pmod{q},$$

with a uniform, e from a *symmetric* error distribution ($-e \sim e$), and Δ the scaling factor. Decryption is $b + as = e + \Delta m$. The noise budget is a monotone function of the level and of $\|e\|$. Note s is small (e.g. ternary) and in general *not* a unit in R_q, so as need not be uniform; the masking below is therefore computational, not information-theoretic (Remark 1).

Decision-RLWE. $(a,\ as + e)$ with a uniform is computationally indistinguishable from (a, u), u uniform [2]. A monitor sees *many* ciphertexts under one key, and the keying material (relinearization/rotation keys) are themselves encryptions involving s; covering this requires the *circular* (key-dependent-message) variant of RLWE on which CKKS already relies. We assume it and use it as a black box.

3 The Key-Free Observer Model

Definition 1 (Key-free observer). *A* key-free observer *is a PPT function $\mathcal{O}$ taking the ciphertext data of the (polynomially many) ciphertexts on the datapath, the public parameters, and the public datapath maps, but* not *the secret s, message m, or noise e. For a predicate π its advantage is $\mathsf{Adv}^{\pi}_{\mathcal{O}} = |\Pr[\mathcal{O} = 1 \mid \pi] - \Pr[\mathcal{O} = 1 \mid \neg\pi]|$. We write $\mathcal{O}^{\mathsf{md}}$ when $\mathcal{O}$ additionally reads the public* metadata *(level, scale, parameter id); the metadata channel is the subject of Section 5.*

This captures any on-die monitor that reads ciphertext words—norm/Hamming-weight gauges, residues against an observer modulus, coefficient moments, or *learned* classifiers (Section 9). The model is deliberately generous: $\mathcal{O}$ sees all ciphertext coefficients and the public computation; it lacks only the key.

4 Region (i): Ciphertext Content Is Masked

Definition 2 (Admissible message/error ensembles). *Fix the public CKKS parameter set, including the ring, modulus chain, scale convention, evaluation-key distribution, and public evaluation schedule. A message/error ensemble $\mathcal{E}$ is* admissible *if the joint distribution of the ciphertexts and evaluation keys generated from $\mathcal{E}$ is covered by the assumed multi-sample circular decision-RLWE pseudorandomness at the chosen parameters. Equivalently, replacing the initial ciphertext and evaluation-key objects by uniform objects of the same public shape is computationally indistinguishable to every PPT key-free observer.*

This definition is deliberately assumption-relative. It excludes artificial conditionings of the error or message distribution that fall outside the RLWE-pseudorandom class assumed for the parameter set. For example, conditioning the error on the predicate "$e = 0$" produces noiseless samples of the form $(b, a) = (-as + \Delta m, a)$, which are not samples from the CKKS error distribution covered by the decision-RLWE assumption. Such a conditioned ensemble is therefore outside the admissible class. The point is not that every inadmissible conditioning immediately gives a simple key-free distinguisher, but that the masking reduction no longer applies; the theorem makes no claim outside the assumed RLWE-pseudorandom regime.

Theorem 1 (Masking). *Assume multi-sample circular decision-RLWE for the CKKS distributions used at the chosen parameters. Let $\mathcal{E}_0$ and $\mathcal{E}_1$ be two admissible plaintext/error ensembles with the same public metadata. Then the corresponding key-free coefficient views are computationally indistinguishable. Consequently, no efficient key-free observer of ciphertext coefficients, without the secret key and without hidden decryption information, distinguishes hidden plaintext content or hidden predicates of the realized within-level RLWE noise with non-negligible advantage, except for predicates whose conditioned ensembles leave the admissible RLWE-pseudorandom class.*

Proof. For a fixed message m, recall the decryption convention $\mathrm{dec}(b, a) = b + a\, s$. Given an RLWE challenge (a, t), where t is either $as + e$ or uniform, construct $\mathrm{ct}' = (\, t + \Delta m,\ -a\,)$. If $t = as + e$, then $\mathrm{dec}(\mathrm{ct}') = (as + e + \Delta m) + (-a)\, s = e + \Delta m$, so ct' is distributed as an honest CKKS ciphertext with error e. If t is uniform, then $t + \Delta m$ is uniform, and ct' is uniform over the same public ciphertext space. Thus a key-free distinguisher between an honest ciphertext and a uniform object would yield an RLWE distinguisher.

The same hybrid argument applies jointly to polynomially many ciphertexts and to the public evaluation keys under the multi-sample circular form of the assumption. By admissibility, the initial objects generated from each $\mathcal{E}_i$ are computationally indistinguishable from uniform objects of the same public shape. Hence the views generated from $\mathcal{E}_0$ and $\mathcal{E}_1$ are both computationally indistinguishable from the same uniform view, and therefore from each other.

For a hidden predicate π of the plaintext or realized within-level noise, apply the preceding argument to the conditional ensembles $\mathcal{E} \mid \pi$ and $\mathcal{E} \mid \neg\pi$, provided both remain admissible. Any non-negligible distinguishing advantage for π would then contradict the assumed RLWE pseudorandomness. □

Corollary 1 (No coefficient noise meter). *For any hidden within-level noise predicate whose conditional distributions remain admissible in the sense above, no efficient key-free observer of ciphertext coefficients has non-negligible correlation with that predicate. A key-free coefficient-magnitude gauge therefore cannot estimate the realized residual RLWE noise; at best it can report public metadata-dependent budget information.*

Why the coefficient magnitude carries no content. The mechanism behind Theorem 1 is the trap practitioners hit. For any modulus p, $b \bmod p = (-as + e + \Delta m) \bmod p$, dominated by the masking term $as \bmod p$. We do *not* claim the residue distributions for different (m, e) are exactly equal (the underlying distribution need not be translation-invariant, so a shift by $\Delta m + e$ is not automatically identity). The claim is computational: any efficient magnitude or residue statistic that correlated with the message or the within-level noise would, by Theorem 1, distinguish the corresponding ciphertexts from uniform and hence yield an RLWE distinguisher. So no such statistic exists, even though the residue distribution is only *computationally* content-independent, not pointwise equal.

Remark 1 (The residue is content-independent, not uniform). Punchline: the residue $b \bmod p$ is *not* uniform—it carries a small, content-free modulo bias—but that bias does not move with the plaintext or the noise, so it gives a key-free monitor no usable signal; hence the right empirical test is distribution-equality, not uniformity. In detail: We deliberately do *not* claim $b \bmod p$ is uniform. Because the working modulus q is not a multiple of an external observer prime p, $b \bmod p$ exhibits a small modulo bias and a goodness-of-fit test rejects uniformity (we observe χ^2 p-values ≈ 0.01 in Section 9). That bias, however, does not give an efficient key-free test for plaintext content or fine within-level noise under the RLWE assumption: any efficient observer that used such residue bias to distinguish admissible plaintext/error regimes would distinguish the corresponding RLWE ciphertext distribution from uniform. An unconditional (information-theoretic) masking would instead require as to be uniform, i.e. s a unit in R_q, which CKKS does not guarantee; hence the masking we state and use is computational (RLWE). The empirical question is therefore not whether residues are perfectly uniform, but whether their distribution changes detectably with the hidden content—so a *distribution-equality* (two-sample) test, not a uniformity test, is the right probe (Section 9).

5 Region (ii): The Coarse Budget Is Public

Theorem 1 forbids reading content from coefficients; it does not forbid a monitor from knowing the noise *budget*, because the deterministic part of the budget is public. Exact CKKS noise/precision depends on more than the level: the public evaluation schedule (rescale history, key-switching/rotation operations, the parameter chain, and value/message bounds), all of which a careful noise estimator tracks [10]. The point is that this dependence is on *public* quantities.

Proposition 1 (Public/residual budget decomposition). *Fix a public CKKS parameter set, a public evaluation schedule, and a public precision/noise estimator* Est *of the kind maintained by an implementation or compiler. Write the estimator as* $\mathsf{Est} = B_{\mathrm{pub}}(\mathsf{md}) + B_{\mathrm{res}}$*, where* md *contains the public level, scale, modulus chain, operation history, rescale schedule, key-switching/rotation schedule, and public value bounds, and where* B_{res} *denotes the remaining dependence on realized encryption and key-switching errors. Then* B_{pub} *is computable key-free from public metadata alone. Hence any budget class defined solely by* B_{pub} *is separable by a key-free monitor with advantage* 1*. By contrast, any hidden predicate of* B_{res} *whose conditional distributions remain admissible is masked from key-free coefficient observers up to negligible advantage.*

Proof. The quantity B_{pub} is, by definition, a deterministic function of public metadata and the public evaluation schedule, so a key-free monitor can compute it exactly. This gives perfect separation for classes defined only by public metadata. The residual term B_{res} depends on realized hidden errors. If a key-free coefficient observer predicted an admissible hidden predicate of B_{res} with non-negligible advantage, then it would predict a hidden predicate of the realized within-level error, contradicting Theorem 1. □

This proposition concerns the implementation's public estimator or conservative bound, not omniscient knowledge of the exact decrypted error. The exact realized precision remains key-dependent except for the public component tracked by metadata.
The right key-free noise gauge is therefore the *public metadata/budget estimator the accelerator already maintains*, not coefficient inspection; the residual that lives *only* in the ciphertext is exactly the part provably unreadable key-free. Corollary 1 is thus a localization, not a counsel of despair: the usable information lives in the public metadata.

6 Region (iii): Public Relations Are Verifiable

The datapath applies fixed, public linear maps (NTT, basis conversion, key-switch inner product), checkable key-free.

Proposition 2 (Key-free relation check; classical). *Let* $M : \mathbb{F}_\kappa^n \to \mathbb{F}_\kappa^m$ *be a fixed public linear map over a prime field* $\mathbb{F}_\kappa$*. Let* x *be the stage input, let* y *be the claimed output, and define the local deviation* $\delta = y - Mx$*. Choose* $r \leftarrow \mathbb{F}_\kappa^m$ *uniformly and independently of* δ*, and precompute* $\rho = M^\top r$*. The test* $\langle r, y\rangle \stackrel{?}{=} \langle \rho, x\rangle$ *accepts every correct computation. If* $\delta \not\equiv 0 \pmod{\kappa}$*, then it accepts a faulty computation with probability exactly* $1/\kappa$*. If* $\delta \equiv 0 \pmod{\kappa}$*, then this particular prime-field check is blind to the deviation.*

Proof. If $y = Mx$, then $\langle r, y\rangle = \langle r, Mx\rangle = \langle M^\top r, x\rangle = \langle \rho, x\rangle$, so correctness is accepted with probability 1. If $y = Mx + \delta$, then the test accepts exactly when $\langle r, \delta\rangle = 0$. For nonzero $\delta \in \mathbb{F}_\kappa^m$, this is a nontrivial linear equation in the

uniform challenge r, and its solution set is a codimension-one affine subspace of $\mathbb{F}_\kappa^m$; it therefore has probability $1/\kappa$ [3]. The independence of r from δ is essential; adaptive faults that choose δ after learning r are outside the proposition. □

Three deployment conditions are worth separating explicitly, because the experiment in §9 sees all three: **(1) algebraic miss**—even for $\delta \not\equiv 0 \bmod \kappa$, the projection can collapse, $\langle r, \delta \rangle = 0$, with probability $1/\kappa$ (the Freivalds bound); **(2) projection blindness**—a fault that is itself $\equiv 0 \bmod \kappa$ is invisible to the test and is *not* covered by the $1/\kappa$ bound (so κ must be large or several primes used to cover R_q faults); and **(3) adaptive faults**—soundness requires the challenge r (equivalently $\rho = M^\top r$) to be *secret from / independent of* whatever induces δ; an adversary who predicts r can craft $\delta \perp r$ and defeat the check. Conditions (1) and (2) together give the $\approx 2/\kappa$ unconditional rate measured in §9.
Proposition 2 is classical, specialized to NTT/lattice hardware in [5,6] and subsumed by verifiable-HE proof systems [7]; we claim no novelty here. We include it because a relation check's output is, by construction, a deterministic function of the public computation, carrying *no* information about s, e, m.

7 The Three-Region Design Space

Design Principle 1 (Where a key-free monitor may act). A key-free monitor of CKKS state operates in three regions: **(i) ciphertext content**—masked, unusable (Thm 1); **(ii) public metadata**—usable, the correct source of the noise budget (Prop 1); **(iii) public relations**—usable, the correct basis of integrity checking (Prop 2). A sound, non-trivial monitor draws on (ii) and/or (iii).

Figure 1 summarizes the three regions and the monitor's admissible inputs; the next section proves the decomposition is exhaustive.

8 Completeness: The Three Regions Are All There Is

Principle 1 would be merely suggestive if some clever observer could extract value from the ciphertext coefficients beyond metadata and relation consistency. We rule this out with a simulation argument (Figure 2).

Execution as a variable graph. Model one run as a graph whose vertices are all ciphertext wires (stage inputs/outputs and the evaluation-key ciphertexts), with public R_q-linear stage maps M_j on the edges and explicit equality/chaining constraints where one stage's output feeds the next. We make four modelling commitments the simulator must honour: (i) the relation family $\mathcal{R}$ is the *complete* set of these stage maps *and* chaining-equality constraints, so a globally consistent assignment respects dataflow, not just per-stage maps; (ii) each stage map M_j is a function of the *public metadata alone*, independent of wire content (true for the CKKS datapath: NTT, basis conversion, and key-switch maps are fixed by

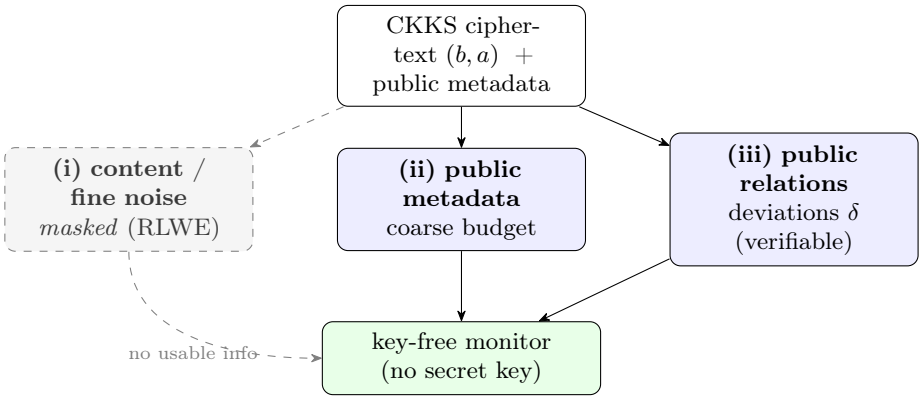

Fig. 1. The three-region decomposition. A key-free monitor sees the ciphertext coefficients and the public metadata but not the secret key. **(i)** The content-bearing part of the coefficients (the message and the fine within-level noise) is computationally masked by RLWE, so the monitor learns nothing usable from it (Theorem 1). **(ii)** The coarse precision budget is determined by public metadata (level, scale, schedule) and is readable key-free (Prop. 1). **(iii)** The public datapath relations can be checked key-free, exposing only the computation deviations δ (Prop. 2). A sound key-free monitor draws only on (ii) and (iii); the completeness theorem (Sec. 8) shows that any remaining coefficient-dependent output is simulable artifact.

public parameters, not by ciphertext values)—this is what makes the propagation the identical public function on the real and simulated sides; (iii) evaluation-key wires are encryptions of s-dependent material and are simulated under the *circular* form of the assumption, not treated as free; (iv) we adopt an *exogenous additive-deviation* fault model—conditional on the public metadata, the deviation vector δ is fixed or sampled *independently of* the hidden plaintexts, the secret key, and the RLWE errors (and hence of the relation-check challenge r). This covers ordinary random hardware faults and injected additive stage faults; it excludes adaptive mechanisms that choose δ as a hidden function of the key, plaintext, or a decryption result.

Definition 3 (Execution transcript). *Fix $\mathcal{R}$, the* complete *family of public R_q-linear stage relations the datapath should satisfy ($y_j = M_j x_j$ for every stage j). The* execution transcript *of a run is the public metadata* md *together with the* residual-deviation *vector $\delta = (\delta_j)_j$, $\delta_j = y_j - M_j x_j$ (so $\delta_j = 0$ iff stage j is fault-free). Here δ_j is the* local *deviation relative to the realized (possibly already-deviated) input x_j of stage j, not the global error relative to a fault-free reference; this is what makes the propagation identical on the real and simulated sides. The transcript records the actual deviations, not merely a pass/fail bit per stage.*

Theorem 2 (Completeness, distributional form). *Assume multi-sample circular decision-RLWE. Fix the complete checked relation family $\mathcal{R}$, consisting*

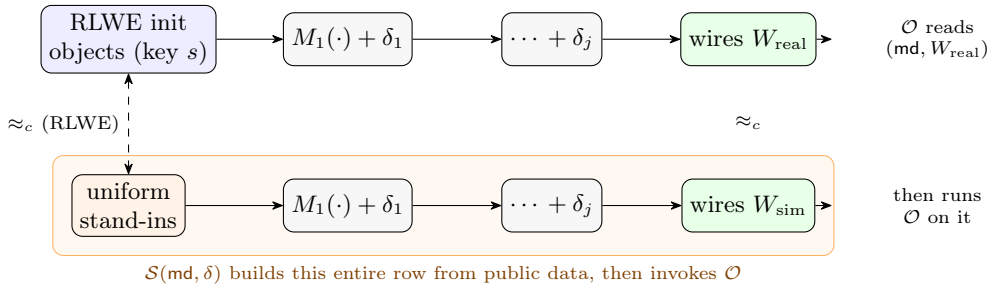

Fig. 2. The completeness simulator. *Top (real run):* the input ciphertexts and evaluation keys are RLWE samples under the secret key s; the public datapath applies the fixed public maps M_j and the additive deviations δ_j to produce the observable wires W_{real}, which the key-free observer $\mathcal{O}$ reads together with the metadata. *Bottom (simulation):* the simulator $\mathcal{S}$ replaces the initial objects with *uniform* stand-ins (indistinguishable from the real ones by RLWE, $\approx_c$) and runs the *same* public maps with the *same* δ. Because the maps depend only on public data, the two wire vectors are computationally indistinguishable, so $\mathcal{S}$ reproduces $\mathcal{O}$'s output from (md, δ) alone—the content-bearing part adds no hidden plaintext or admissible fine-noise information (Theorem 2).

of the public linear stage maps and the public chaining/equality constraints of the datapath, and assume the wire vector is efficiently sampleable from the initial objects and (md, δ) *(which holds for the linear CKKS datapath). Assume the* exogenous additive-deviation model*: the deviation transcript* $\delta = (\delta_j)_j$*,* $\delta_j = y_j - M_j x_j$*, is, conditioned on* md*, fixed or sampled independently of the hidden plaintexts, the secret key, the RLWE errors, and the relation-check challenge* r*. Then for every PPT key-free observer* $\mathcal{O}$*, there exists a PPT simulator* $\mathcal{S}^{\mathcal{O}}$ *such that, given only the public metadata* md *and the local deviation transcript* δ*, the simulator's output distribution is computationally indistinguishable from the real output distribution of* $\mathcal{O}$ *on the coefficient wires:*

$$\mathcal{O}(\mathsf{md}, W_{\text{real}}) \;\approx_c\; \mathcal{S}^{\mathcal{O}}(\mathsf{md}, \delta).$$

Thus the coefficients may still supply pseudorandom-looking bits to $\mathcal{O}$*, but those bits are simulable from fresh randomness and carry no additional information about hidden plaintexts, secrets, or admissible residual-noise predicates beyond what is already present in* (md, δ)*.*

Proof. Let W_{real} be the vector of all coefficient-state wires in the real execution. The initial ciphertexts and evaluation-key objects are sampled from the CKKS/RLWE distribution under the secret key s. The public datapath then applies the public maps M_j in topological order, adding the local exogenous deviation δ_j at stage j: $y_j = M_j x_j + \delta_j$.

Define $\mathcal{S}^{\mathcal{O}}$ as follows. Given only (md, δ), the simulator samples uniform stand-ins of the same public shape as the initial ciphertexts and evaluation-key objects. It then runs the same public datapath, in the same topological order,

adding the same local deviations δ_j, obtaining a simulated wire vector W_{sim}. Finally, it runs $\mathcal{O}(\mathsf{md}, W_{\mathrm{sim}})$ and outputs whatever $\mathcal{O}$ outputs.

By the multi-sample circular decision-RLWE assumption, the real initial ciphertext and evaluation-key objects are computationally indistinguishable from the uniform stand-ins of the same public shape. The transformation from the initial objects to the full wire vector is a public polynomial-time map once (md, δ) is fixed. This is where the exogenous model is used: the deviations are not chosen as hidden functions of the secret key, plaintexts, RLWE errors, or the challenge randomness of the checks. Computational indistinguishability is preserved under public polynomial-time transformations, so $(\mathsf{md}, W_{\mathrm{real}}) \approx_c (\mathsf{md}, W_{\mathrm{sim}})$. Applying the PPT observer $\mathcal{O}$ preserves computational indistinguishability. Therefore $\mathcal{O}(\mathsf{md}, W_{\mathrm{real}}) \approx_c \mathcal{O}(\mathsf{md}, W_{\mathrm{sim}}) = \mathcal{S}^{\mathcal{O}}(\mathsf{md}, \delta)$. The same argument covers reconvergent and chained wiring because both the real and simulated executions use the identical public propagation rules and the same local deviation transcript. □

Corollary 2 (Sharp design rule). *Up to negligible terms, a key-free CKKS monitor's useful information is distributionally simulable from the public metadata and the local computation-deviation transcript. The content-bearing part of the ciphertext coefficients contributes no hidden plaintext or admissible fine-noise information. The monitor may still output pseudorandom-looking functions of the coefficients, but such outputs are simulable artifacts, not actionable information about the encrypted computation.*

Remark 2 (Why the transcript carries δ, not merely pass/fail bits). Giving the simulator only a Boolean pass/fail vector would be insufficient: a key-free observer can read the faulty coefficients and therefore may depend on the actual deviation value. The correct ideal transcript is the local deviation δ itself. Actual relation checks may reveal only a projection or pass/fail evidence about δ (Proposition 2 reads a random projection of δ); the theorem gives the stronger boundary that even full access to δ, plus public metadata, suffices to simulate every key-free observer's output distribution.

Scope and relation to semantic security. The fault-free case ($\delta = 0$) of Theorem 2 is the simulation-paradigm consequence of CKKS semantic security (pseudorandom ciphertexts are simulable from public data); our contribution is the *faulty* case—formalizing that the only thing a key-free monitor learns beyond metadata is the deviation δ, and that δ, not a pass/fail bit, is the right transcript. The statement is relative to the checked family $\mathcal{R}$: a deviation outside $\mathcal{R}$ is, by definition, not in δ and is undetectable key-free (consistently for $\mathcal{O}$ and $\mathcal{S}$). The efficient-sampling hypothesis holds for the linear datapath relations; we do not claim it for arbitrary nonlinear predicates. Within these bounds the three-region picture is a theorem, not a heuristic.

9 Experiments

The experiments are small, reproducible *sanity checks* consistent with the model, not a security validation—the security rests on the reduction (Theorem 1),

and the toy ring sizes below are far from production parameters. We instantiate R_q with a 60-bit word prime, ternary s, $\Delta = 2^{40}$, and a symmetric discrete-Gaussian error. The scripts, fixed seeds, feature definitions, train/test splits, and exact commands are released with the paper (`exp/observe.py`, `exp/learned_observer.py`, `exp/settle.py`; pure NumPy). The advantage estimator fits a threshold/classifier on a training half and evaluates on a disjoint half (held-out, unbiased); we always report a **null baseline**—the advantage the same estimator returns between two *identical* classes—so "statistical zero" is calibrated against finite-sample bias rather than asserted, and we treat any advantage within $\approx 2\sigma$ of the null as chance.

(1) Content is masked, even against a learned observer. Beyond hand-built statistics we train a logistic-regression observer on a ∼35-dimensional key-free feature vector (residues against five observer primes; centred-limb moments; a coarse coefficient histogram), a representative key-free monitor. Held-out advantage for distinguishing within-level noise (σ=3.2 vs. 3200) and message (m=0 vs. 1):

Ring N	n/class	null floor	noise advantage	message advantage
256	600	0.067	0.023	0.003
512	600	0.033	0.007	0.017
1024	500	0.027†	0.007	0.064†

Table 1. Learned key-free observer, held-out advantage (0 =chance). Noise advantage is at or near the null floor for every ring size, as is message advantage for every cell except the footnoted N=1024 message cell, which sits $\approx 2\sigma$ from chance; even an optimized classifier reads no content beyond the floor. (†) The N=1024 row is reported as the mean over 4 seeds, with the null floor estimated by a label-permutation test: message advantage 0.064 ± 0.012 vs. permutation null 0.027 ± 0.017 (i.e. within $\approx 2\sigma$ of chance). An earlier single n=300 run gave an inflated 0.120; the seeded permutation estimate resolves it as small-sample noise.

(2) Hand-built statistics, with-key control, and the budget. At $N = 512$, $n =$ 1000/class, held-out: residue-mod-p and var-of-limb give net signal ≈ 0 for message and ≤ 0.04 for within-level noise (at the null floor); the *same* magnitude statistic computed after a partial decryption (which uses s) reaches 0.998—the information exists but is key-gated. The *public level metadata* separates budget classes at advantage 1.000 while the coefficients do not: the coarse/fine split of Proposition 1.

(3) Uniformity vs. content-independence. A χ^2 goodness-of-fit on $b \bmod p$ rejects uniformity ($p \approx 0.01$, Remark 1)—the expected modulo bias—yet the two-sample/learned tests above show the distribution does not depend on (m, e). Uniformity is the wrong probe; distribution-equality is the right one, and content is masked under it.

(4) Public relations are verifiable key-free. Over a prime $\kappa = 1009$ with a *precomputed* projection (no map recomputation), 10,003 injected single-coefficient faults gave a false-accept (missed-fault) rate of 0.21% (95% CI ±0.09%). This matches the prediction $\approx 2/\kappa$, not $1/\kappa$: a missed fault occurs when either $\langle r, \delta \rangle \equiv 0$ (probability $1/\kappa$) *or* the injected deviation is itself $\equiv 0 \bmod \kappa$ (another $\approx 1/\kappa$; measured 0.05%), the latter being a fault aligned to the tested prime as flagged after Prop. 2. A word-size prime drives both to $\approx 2^{-60}$. Detection was otherwise near-complete. This realizes region (iii) recomputation-free; the earlier informal "0.5%" figure was a 5/1000 small-sample estimate, superseded here.

10 Implications for FHE Hardware Monitors

- **Gauge the budget from metadata, not coefficients.** A bootstrap trigger should read the public level/scale counter (region ii); inspecting ciphertext magnitudes (region i) is provably uninformative (Cor. 2).
- **Spend integrity effort on relation checks.** The sound key-free integrity class is relation consistency (region iii), realizable recomputation-free via a precomputed projection, revealing nothing about content.
- **Integrity is not confidentiality.** The results concern the *values* a monitor reads, not the power/timing side channels of the circuit that reads them.

11 Related Work

CKKS security reduces to Ring-LWE [1,2]; our masking statement (Thm 1) is the monitor-facing form of that pseudorandomness. The fault-free case of our completeness theorem is the standard simulation-paradigm consequence of semantic security; the contribution is the *faulty*-case refinement (Remark 2): a key-free monitor's output distribution is simulable from metadata plus the computation-deviation transcript δ. We are careful to position this as a refinement of semantic security, not a new hardness result. Our model is also strictly weaker than the decryption-oracle/IND-CPA$^{\mathrm{D}}$ setting [8] in which key-recovery attacks on approximate HE operate [9]: those use the rounded *decryption* output, which a key-free monitor never sees; the existence of such attacks underscores why the no-decryption-oracle boundary of our model matters. Accurate noise-budget estimation [10] is a key-holder/parameter tool; our coarse/fine split says which part a *key-free* monitor may see. Region (iii) is furnished by Freivalds [3], ABFT [4], NTT-specific error detection [5,6], and verifiable/attested HE [7,11]; we attribute the positive side to them. We are not aware of a prior statement of the coarse/fine key-free observability decomposition or the deviation-transcript completeness bound for key-free CKKS monitoring.

FHE hardware accelerators. Our monitor model sits on top of the CKKS datapath that recent accelerators implement—F1 [12], CraterLake [13], BTS [14], and the broader design space surveyed in [15]—whose NTT, basis-conversion, and key-switch stages are exactly the public linear maps that region (iii) checks. We are

agnostic to the specific accelerator and ask what *any* key-free on-die monitor over such a datapath can learn; prior hardware-integrity work in this space targets NTT-stage error detection [5,6], and we frame the complementary observability question.

12 Limitations and Conclusion

What this does not cover. The results are computational and assumption-relative. They apply to key-free *coefficient* observers under multi-sample circular decision-RLWE, and to faults captured by the *exogenous additive-deviation* model. They do not cover key holders; decryption / IND-CPA$^{\mathrm{D}}$ oracles; physical side channels such as timing/power/EM leakage of the circuit that reads the values; *adaptive* faults that depend on hidden plaintexts, secrets, errors, or relation-check challenges; *content-dependent* hardware faults (value-dependent bit-flips, stuck-at, saturation, carry, timing, or memory-aliasing faults whose effective δ is a function of the wire value); faults whose effect is outside the checked relation family; and arbitrary *nonlinear* predicates for which efficient relation-consistent sampling is unavailable. The budget statement concerns public estimators and conservative metadata-derived bounds; it is not a claim that the exact realized CKKS error is public. These are the boundaries of the model, stated so the title's "hardware monitors" is not read more broadly than proved.

The masking and completeness results are computational (circular RLWE) and concern observers without the key; they say nothing to a key holder. Completeness is relative to the checked relation family and assumes efficient sampling of relation-consistent coefficients (which holds for the linear CKKS datapath). The empirics use small rings for reproducibility; the masking argument is parameter-independent via the reduction, and the multi-N sweep is consistent with that. Within these bounds the picture is clean and useful: a key-free monitor should read the budget from public metadata and check public relations; it cannot, and should not try to, read content from the ciphertext—and, by Theorem 2, the remaining coefficient-dependent output is simulable pseudorandom artifact, not hidden plaintext or admissible fine-noise information.

Use of AI assistance. A large-language-model assistant was used to help draft and revise the prose of this paper and to implement and run the accompanying experiment scripts. All definitions, theorem statements, proofs, and empirical claims were checked and verified by the authors, who take full responsibility for the content.

References

1. J. H. Cheon, A. Kim, M. Kim, Y. Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. ASIACRYPT 2017. IACR ePrint 2016/421.
2. V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning With Errors Over Rings. EUROCRYPT 2010.

3. R. Freivalds. Probabilistic Machines Can Use Less Running Time. IFIP Congress, 1977.
4. K.-H. Huang, J. A. Abraham. Algorithm-Based Fault Tolerance for Matrix Operations. IEEE Trans. Computers, 1984.
5. M. Abdelmonem, L. Holzbaur, H. Raddum, A. Zeh. Efficient Error Detection Methods for the Number Theoretic Transforms in Lattice-Based Algorithms. IACR ePrint 2025/170; CASCADE 2025.
6. K. Ahmadi, S. Aghapour, M. Mozaffari Kermani, R. Azarderakhsh. Efficient Algorithm Level Error Detection for Number-Theoretic Transform used for Kyber Assessed on FPGAs and ARM. arXiv:2403.01215, 2024.
7. I. Cascudo, A. Costache, D. Cozzo, D. Fiore, A. Guimarães, E. Soria-Vazquez. Verifiable Computation for Approximate Homomorphic Encryption Schemes. CRYPTO 2025; IACR ePrint 2025/286.
8. B. Li, D. Micciancio. On the Security of Homomorphic Encryption on Approximate Numbers. EUROCRYPT 2021; IACR ePrint 2020/1533.
9. Q. Guo, D. Nabokov, E. Suvanto, T. Johansson. Key Recovery Attacks on Approximate Homomorphic Encryption with Non-Worst-Case Noise Flooding Countermeasures. USENIX Security 2024.
10. J.-P. Bossuat, A. Costache, C. Mouchet, L. Nürnberger, J. R. Troncoso-Pastoriza. Accurate and Composable Noise Estimates for CKKS with Application to Exact HE Computation. IACR Communications in Cryptology, vol. 2, no. 2, art. 8, 2025.
11. M. H. Santriaji, J. Xue, Y. Zhang, Q. Lou, Y. Solihin. DataSeal: Ensuring the Verifiability of Private Computation on Encrypted Data. IEEE Symposium on Security and Privacy 2025, pp. 2378–2394; arXiv:2410.15215.
12. A. Feldmann, N. Samardzic, A. Krastev, S. Devadas, R. Dreslinski, K. Eldefrawy, N. Genise, C. Peikert, D. Sanchez. F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption. MICRO 2021.
13. N. Samardzic, A. Feldmann, A. Krastev, N. Manohar, N. Genise, S. Devadas, K. Eldefrawy, C. Peikert, D. Sanchez. CraterLake: A Hardware Accelerator for Efficient Unbounded Computation on Encrypted Data. ISCA 2022.
14. S. Kim, J. Kim, M. J. Kim, W. Jung, M. Rhu, J. Kim, J. H. Ahn. BTS: An Accelerator for Bootstrappable Fully Homomorphic Encryption. ISCA 2022.
15. J. Zhang, X. Cheng, L. Yang, J. Hu, X. Liu, K. Chen. SoK: Fully Homomorphic Encryption Accelerators. ACM Computing Surveys, 56(12), Article 316, 2024; arXiv:2212.01713.