

LATTICEBLINDFOLD: A LATTICE-BASED ANALOGUE OF NOVABLINDFOLD

LUCA DALL’AVA

ICME Labs
<https://www.icme.io>

ABSTRACT. Folding schemes compress many instances of a relation into a single accumulated one and, via composition with the Fiat–Shamir heuristic, yield SNARKs for arbitrarily large computations. However, essentially every folding scheme beyond Nova itself (including the lattice-based SuperNeo [NS26], LatticeFold(+) [BC24, BC25], and Cyclo [GLLO26]) is only *randomizing*, not *blinding* (i.e. honest-verifier zero-knowledge): its folding transcript leaks information about the witnesses being folded. We present LatticeBlindFold, a first lattice-based, plausibly post-quantum-secure analogue of the NovaBlindFold protocol [KS23, KS25], obtained by making SuperNeo blinding. This is a first step, intended to establish feasibility; for simplicity we restrict to the interactive setting. We also hope the note serves as a record of the difficulties one encounters in achieving zero-knowledge for lattice-based folding schemes.

Our central technical device is the ABDLOP commitment scheme [BDL⁺16, LNP22], used as a commit-and-prove backbone. We mask SuperNeo’s Sum-Check transcript via Libra-style polynomial masking, replace its plaintext evaluation hints with ABDLOP commitments checked homomorphically, and employ rejection sampling so that the randomized folded instance-witness pair, salts included, is simulatable. Since ABDLOP is only known to be secure over the base ring R_F and not over the extension ring R_K that SuperNeo’s Sum-Check runs over, we give a component-wise instantiation of ABDLOP over R_K , translating every relation into a pair of R_F -relations once, at the level of public parameters. The resulting protocol is complete, knowledge-sound, and blinding (when a single fresh R1CS instance is folded), with security reducing to Module-SIS/Module-LWE assumptions over cyclotomic rings (together with the Extended-MLWE variant of [LNS20]). Completeness moves from perfect completeness to a statistical one, due to the introduction of rejection sampling. The price of blinding is paid through the parameters rather than the asymptotics. Rejection sampling forces a norm-decomposition depth $k = \Theta(\log n_F)$, whereas SuperNeo needs only $k = \Theta(1)$. We record as a corollary an accumulator-free variant, dropping the prover-sampled blinding pairs that the main protocol carries for interface compatibility with Π_{SuperNeo} , which saves a few decomposition digits ($k = 26$ rather than 31 at our parameters) at no cost in blinding. At a matched k the two schemes agree in prover time, verifier time, and communication up to constant factors, but measured against SuperNeo at its native parameters LatticeBlindFold carries a $\Theta(\log n_F)$ multiplicative overhead in all three. Once again, all security figures we quote are interactive; a Fiat–Shamir transform instantiation would require a higher degree fields while here we focus on degree 2 for simplicity of exposition. LatticeBlindFold inherits SuperNeo’s compatibility with small-field arithmetic and offers a modular, plausibly post-quantum-safe route to introducing zero-knowledge on top of a SNARK. We stress the shape of what we prove here: Π_{LBF} is a *single* interactive step, taking its input uncommitted and outputting k committed evaluation claims together with the ABDLOP openings certifying them. Turning it into a deployable wrapper requires a decider for those claims, whether by having a downstream verifier consume them directly or by arithmetizing the ABDLOP verification; we do not construct one here, and neither recursive composition nor the instance-in/instance-out folding interface is claimed. This work is directly motivated by the *Jolt Atlas* zkML framework [BCDG26].

CONTENTS

1. Introduction	2
1.1. Motivation	2
1.2. LatticeBlindFold	4
1.2.1. Applications	5
1.3. Our contributions	5
1.4. Paper organization	8

2. Recalls and preliminaries	8
2.1. Notation and Global Parameters	8
2.2. Interactive reductions	10
2.3. Strong Sampling Sets: Controlling Norm Growth in Folding	14
2.4. Recalls of SuperNeo	15
2.4.1. The embedding and the relations	15
2.4.2. The interactive reductions	16
2.5. Ring commitments	17
2.5.1. Ajtai Commitment Schemes	18
2.5.2. The ABDLOP commitment scheme	18
2.6. Security assumptions	19
3. The zk-fying components	20
3.1. Hiding for Ajtai commitments	20
3.2. Companion procedures from the ABDLOP proofs of knowledge	21
3.2.1. Vanishing constant coefficients	23
3.3. Instantiating ABDLOP over R_K Componentwise	27
3.3.1. The Rank-Doubling Embedding	27
3.3.2. Linear Relations	27
3.3.3. Quadratic Relations and Automorphisms	28
3.3.4. Norms	29
3.3.5. Soundness of K -Valued Challenge Masking	29
3.3.6. Complexity	30
3.4. Libra-style masking	30
3.5. Replacing the PCS with the ABDLOP commitment scheme	31
3.6. A brief recall on rejection sampling	32
3.7. The relations	33
4. LatticeBlindFold	36
4.1. The building blocks of LatticeBlindFold	36
4.1.1. First Reduction: R1CS Reduction	36
4.1.2. Second Reduction: Random Linear Combination	41
4.1.3. Third Reduction: Decomposition Reduction	43
4.2. The LatticeBlindFold protocol	44
4.2.1. A blueprint for LatticeBlindFold as a zero-knowledge folding scheme	46
4.3. Main results	47
4.3.1. The consolidated completeness error	49
4.3.2. The consolidated knowledge-soundness error	49
4.3.3. Efficiency measures	51
4.3.4. Choice of parameters	53
4.4. The accumulator-free variant	57
4.5. A comment about Fiat-Shamir	58
5. Future work and open research directions	59
References	60
Appendix A. AI Usage Disclosure	62
Appendix B. Deferred proofs	62
B.1. Proof of Lemma 4.9	62
B.2. Proof of Lemma 4.10	64
B.3. Proof of Lemma 4.12	69
B.4. Proof of Theorem 4.13	71

1. INTRODUCTION

1.1. Motivation. The intersection of zero-knowledge (ZK) proofs and succinct argument systems is a central and rapidly evolving focus of modern cryptography: enabling one party to convince another that a

computation was performed correctly, without revealing the underlying data, while keeping the proof itself cheap to produce and to verify. A particularly fruitful framework for this line of work is that of *folding schemes*, as introduced by Nova [KST22, KS23]. Folding schemes compress many instances of a relation into a single accumulated instance which can then be checked; if this accumulated instance is verified, then also the instances we started with are (up to negligible probability) verified. Interestingly, repeated application of a folding step yields (informally and with composition with the Fiat–Shamir heuristic) a SNARK: the prover reduces an arbitrarily large computation to a single, efficiently checkable instance. Nova and its descendants (HyperNova [KS23], SuperNeo [NS26]) all follow this paradigm. Furthermore, the Nova scheme, in its NovaBlindFold protocol instantiation (as formalized in [KS23, KS25]) provides a systematic way of introducing zero-knowledge into this process, without stacking a separate ZK argument system on top of an already-existing SNARK. This is the scheme that we generalize here to the lattice-based framework.

This second application of folding schemes is an active line of research; recent efforts to add zero-knowledge on top of existing SNARKs include NovaBlindFold [KS23, KS25] and VEIL [DHR26].

Our own interest in this question is concrete rather than purely theoretical. At ICME Labs we develop *Jolt Atlas* [BCDG26], a zero-knowledge machine learning (zkML) framework built on top of the Jolt zkVM [AST23], in which zero-knowledge is currently obtained via the NovaBlindFold trick recalled below. Let us remark that by zero-knowledge we always mean *honest-verifier* zero-knowledge (HVZK), and we adopt this convention throughout. The increasingly pressing need for a post-quantum-safe transition, together with the computational benefits of working over small fields, led us to ask whether the same blinding trick could be transplanted from the elliptic-curve setting of Nova to a lattice-based, SuperNeo-style architecture. Answering this question is the starting point of the present paper, and we return to this motivating application in §1.2.1 once the relevant background is in place.

With the notable exception of Nova itself, however, existing folding schemes do not achieve zero-knowledge out of the box. The transcript produced during a folding step leaks information about the witnesses being folded. NovaBlindFold addresses the following natural question:

Can we obtain a blinding (i.e. zero-knowledge) folding scheme without introducing a separate ZK argument layer on top?

The answer is yes, and the mechanism is remarkably simple. Recall the following informal definition, which we make precise in Definition 2.11 below.

Definition (Blinding (Informal)). *A folding reduction $\Pi: \mathcal{R} \times \mathcal{R}' \rightarrow \mathcal{R}$ is blinding if there exist a sampling algorithm `sample` and a simulator $\mathcal{S}$ such that the interaction transcript is computationally indistinguishable from a simulated one, given only the instances (never the witnesses).*

Concretely, blinding for the Nova protocol is achieved as follows:

- (1) The prover samples a randomized instance-witness pair $(u_r, w_r) \leftarrow \text{sample}$, independently of the real witness (u, w) it wishes to hide.
- (2) It folds the two pairs $((u_r, w_r), (u, w))$ together; by linearity of the folding operation, the output witness is $w_{\text{out}} = \rho \cdot w_r + w$ for a verifier challenge ρ .
- (3) Because w_r is freshly random and ρ is a verifier challenge unknown to the prover at sampling time, w_{out} is computationally indistinguishable from a fresh sample of `sample`.

The prover can therefore transmit w_{out} *in the clear*. No hiding commitment is required, since the randomization is already baked into the folding step itself. This yields zero-knowledge on top of essentially any SNARK, provided the underlying folding scheme satisfies the blinding property, i.e. that the output instance-witness distribution is independent of the real witness whenever one of the two inputs being folded is freshly random.

Remark. *The Nova folding scheme is blinding. HyperNova, however, satisfies only a strictly weaker property, called randomizing: the folded instance-witness pair is indistinguishable from a randomly sampled pair, but this alone does not imply zero-knowledge.*

This difference leaves a crucial gap. Almost all widely used lattice-based folding schemes, including LatticeFold [BC24], LatticeFold+ [BC25], SuperNeo [NS26], and the more recent Cyclo [GLLO26], are generalizations of HyperNova rather than of Nova, and consequently inherit the same obstruction: they are sound and complete, but not blinding.

The one lattice-based exception we are aware of is Lova [FKNP24], a direct adaptation of Nova whose folding step is linear in exactly the sense the blinding trick requires; it does not, however, achieve zero-knowledge, and it builds on unstructured SIS over $\mathbb{Z}/2^{64}\mathbb{Z}$ rather than on the module setting. We take the SuperNeo route instead: our motivating application (§1.2.1) is Sum-Check-based and operates over small fields, and SuperNeo’s reduction is Sum-Check-native over $R_{\mathbb{F}} = \mathbb{F}[X]/(\Phi(X))$, which lets us plug Π_{LBF} in without re-encoding the constraint system, whereas $\mathbb{Z}/2^{64}\mathbb{Z}$ is not an integral domain and carries no multilinear extension to blind. The ABDLOP commitment scheme [LNP22] and its ZKPoK, on which all of §4 rests, are module-lattice objects, so transplanting them to the Lova setting would mean re-deriving the whole commit-and-prove layer. A zero-knowledge Lova would be a genuinely different point in the design space, and we regard it as an interesting open question.

Two further recent lattice-based folding schemes are SALSAA [KLOT25], whose folding application operates natively on ℓ_2 -norm-bounded witnesses (and whose exact norm-check is of independent interest here, cf. Remark 4.8), and PikkuFold [Osa26], which also highlights the necessity of obtaining zero-knowledge folding schemes. Neither claims a blinding property.

Throughout the paper, we focus on SuperNeo [NS26] as it is the protocol we build on; however, we believe that our modifications can be applied to similar Sum-Check-based folding schemes. SuperNeo is a folding scheme for the norm-bounded CCS relation $\text{CCS}(b, \mathcal{L})$ ([NS26, Definition 19], recalled as Definition 2.21 below) over the cyclotomic ring $R_{\mathbb{F}} = \mathbb{F}[X]/(\Phi(X))$, where $\mathbb{F}$ is a small field and Φ a cyclotomic polynomial. It achieves knowledge soundness and completeness, but its blinding property fails at exactly two points:

- (1) The Sum-Check protocol: SuperNeo’s core reduction relies on the Sum-Check protocol to verify that the constraint polynomial vanishes on the Boolean hypercube. The transcript of a (plain) Sum-Check execution reveals partial evaluations of this polynomial at each round, which encode information about the underlying witness.
- (2) The evaluation hints: At the end of the Sum-Check phase, the prover sends hint values

$$y'_{i,j} = \widetilde{M_j z_i}(r') \in R_{\mathbb{K}}, \quad \forall i \in [K+k], \forall j \in [t],$$

for a suitable extension field $\mathbb{K}/\mathbb{F}$. These are linear functions of the witnesses z_i evaluated at a random point $r' \in \mathbb{K}^{\log_2(m)}$, and are sent to the verifier *in the clear*.

Neither leak is fatal in a single execution, but an honest-but-curious verifier who accumulates these quantities across many folding rounds can recover substantial structural information about the witnesses. That is precisely the situation that NovaBlindFold was designed to rule out in the elliptic-curve setting, and that we rule out here in the lattice setting.

Let us stress that a recurring theme throughout our construction is that every quantity that would otherwise leak information is routed through the ABDLOP commitment scheme [BDL⁺16, LNP22]. Unlike the succinct Ajtai commitment SuperNeo already relies on, ABDLOP is non-succinct but computationally hiding and comes equipped with an efficient zero-knowledge proof of knowledge, which lets us replace every point where SuperNeo discloses a value in the clear with a commit-and-prove step instead.

1.2. LatticeBlindFold. The goal of this paper is to close this gap. We construct LatticeBlindFold, a blinding procedure based on a folding scheme over lattices, i.e. the lattice-based analogue of NovaBlindFold. Let us remark that, to the best of our knowledge no lattice-based folding procedure has been shown to satisfy a blinding property in the sense of Definition 2.11. Beyond recovering zero-knowledge, working over lattices rather than elliptic curves brings two further, largely independent, advantages, which we discuss in turn.

- **Small-field arithmetic:** SuperNeo operates over fields $\mathbb{F}$ of roughly 61–64 bits and quadratic extensions $\mathbb{K}$ of roughly 122–128 bits, rather than the 254-bit prime-order field required by elliptic-curve-based SNARKs such as Nova. Constraint systems that are naturally expressed over small fields (as is typical of zkVM lookup arguments) benefit directly from this.
- **Plausible post-quantum security:** Every hardness assumption we rely on reduces to Ring-SIS/Module-SIS over $R_{\mathbb{F}}$, believed to be hard even against quantum adversaries. This is in sharp contrast to NovaBlindFold, whose security rests on the hardness of the discrete logarithm problem, a problem that Shor’s algorithm breaks in polynomial time on a quantum computer. This concern is not merely theoretical; see e.g. [BZG⁺26] on the concrete quantum risk to elliptic-curve-based cryptography.

These advantages come at a cost, though a subtler one than might be expected. None of the machinery blinding requires (the masked Sum-Check, the ABDLOP commitments to the evaluation hints, the batched anchoring proofs of knowledge, the degree-0 surrogates and their constant-coefficient proofs) changes the order of prover time, verifier time or communication at a fixed decomposition parameter k . The additive overhead is the public data of the instances the prover introduces, namely, the mask’s public instance and that of the k sampled blinding pairs, of size $n_{\mathbb{F},\text{in}}$ each (the k pairs share a single one, since split_b and $\mathcal{L}_{\text{in}}$ commute), together with the k compact Ajtai commitments c_i^{blind} and the kt evaluation hints those pairs carry. The last two are $\Theta(k)$, hence of the same order as the k output instances that any decomposition-based folding step already transmits. They appear in the constant on the $\log_2(m)$ term of Table 1, not its order, but they are real bytes and are charged as such. The price is paid through k itself. Rejection sampling inflates the norm bound by a factor $\sqrt{n_{\mathbb{F}}}$, forcing $k = \Theta(\log n_{\mathbb{F}})$ where Π_{SuperNeo} may take $k = \Theta(1)$. Therefore, when each protocol is read at its own parameters, Π_{LBF} costs a $\Theta(\log n_{\mathbb{F}})$ multiplicative factor in all three metrics, and this in turn feeds into the concrete security estimates. We report both schemes at a matched k in Table 1 so that the two columns describe the same protocol structure and can be read metric by metric, but the matched- k reading understates the true gap and should not be taken as the operative comparison. We make this precise in §4.3.3, and in particular in Remark 4.18.

1.2.1. Applications. As mentioned above, our primary motivation for developing LatticeBlindFold is to provide a modular, plausibly post-quantum-safe way to introduce zero-knowledge on top of the *Jolt Atlas* architecture [BCDG26], a zero-knowledge machine learning (zkML) framework that extends the Jolt zkVM [AST23] (and proving system) to model inference, developed by ICME Labs. Jolt Atlas currently obtains zero-knowledge via the NovaBlindFold trick; the increasingly pressing need for a post-quantum-safe transition, together with the computational benefits of small-field arithmetic discussed above, motivated us to seek a lattice-based analogue of this trick that could be plugged into a SuperNeo-based (rather than Nova-based) architecture.

We stress, however, that the relevance of this construction extends well beyond Jolt Atlas specifically. The modularity of the blinding-folding paradigm, namely the fact that zero-knowledge can be bolted onto an existing (non-ZK) SNARK without redesigning it from scratch, is a central topic for modern cryptographic applications generally. Any system built on a HyperNova-style folding scheme, lattice-based or not, faces the same witness-leakage obstruction we describe above, and stands to benefit from the same kind of modular fix. This direction is not idiosyncratic to our own work: a growing number of teams are exploring lattice-based zkVM architectures, and LatticeBlindFold is designed to fit naturally within that broader shift toward plausible post-quantum, modular proof systems, rather than being tailored narrowly to a single application.

1.3. Our contributions. We summarize here the main steps and contributions of this work; the technical terms below are all defined precisely in the body of the paper, and a reader mainly interested in the high-level picture can safely skip ahead to §1.4.

- We introduce a norm-bounded *committed* CCS evaluation relation $\text{CE}_{\text{com}}(b, \mathcal{L}, \text{Commit})$ (Definition 3.21), its restricted version (Definition 3.25), as well as a Relaxed committed CCS evaluation relation which augment SuperNeo’s evaluation relation with ABDLOP commitments to the evaluation hints and their salts (Definition 3.22), a relaxed committed CCS evaluation relation for intermediate constructions (Definition 3.24), and padded/unpadded layouts (Definition 3.28). The reductions can then check consistency without ever seeing the hints in the clear; the restricted/unpadded relations are what makes zero-knowledge attainable. Due to all these modifications, at the same time, we adjust the setting of interactive reductions introduced in [NS26] to our needs.
- Even though we keep considering the compact Ajtai commitment scheme, we need to introduce an extra blinding block in order to achieve (statistical) hiding (Definition 3.1, Lemma 3.3) alongside the relaxed binding property (Definition 2.27).
- We mask the Sum-Check transcript with a Libra-style polynomial $p(\vec{X}) = a_0 + \sum_{i=1}^{\ell} \tilde{p}_i(X_i)$ [XZZ⁺19]. Unlike loc. cit., whose verifier reconstructs the unmasked evaluation $Q(\vec{r}')$ once the mask’s own evaluation is opened by the polynomial commitment scheme (Remark 3.18.(1)), we reveal neither: the final check takes place entirely at the level of the commitments, so $p(\vec{r}')$ is never opened and $Q(\vec{r}')$ never formed as a field element in $\mathcal{V}$ ’s view. The complete inventory of what leaves the prover in the clear is §4.1.1.1.

- We replace SuperNeo’s plaintext evaluation hints with a commit-and-prove step using the ABDLOP commitment scheme [BDL⁺16, LNP22], and show that equality checks on the hints can instead be performed homomorphically at the level of the commitments.
- Since ABDLOP and its zero-knowledge proof of knowledge [LNP22] are only known to be secure natively over $\mathbb{R}_{\mathbb{F}}$, while SuperNeo’s Sum-Check transcript lives over $\mathbb{R}_{\mathbb{K}}$, we cannot invoke them over $\mathbb{R}_{\mathbb{K}}$ off the shelf. We give instead a *component-wise* instantiation (§3.3): writing $\mathbb{K} = \mathbb{F}[Y]/(Y^2 - \nu)$ for a quadratic non-residue ν , we commit to the $\{1, Y\}$ -coordinates of $\mathbb{R}_{\mathbb{K}}$ -elements directly as $\mathbb{R}_{\mathbb{F}}$ -elements and translate every $\mathbb{R}_{\mathbb{K}}$ -relation into an equivalent pair of $\mathbb{R}_{\mathbb{F}}$ -relations once, at the level of public parameters, before invoking [LNP22]’s proof system unchanged.
- Several equalities we must certify are not identities in $\mathbb{R}_{\mathbb{K}}$ but constrain only the constant coefficient of a committed ring element: this is the case for the degree-0 surrogates $\hat{y}_{i,j}$, which force the committed hints into the base field before the degree-two R1CS check, and for the anchoring proofs binding the Sum-Check output to the committed values. Since [LNP22]’s system certifies full-ring linear relations $\mathbf{u} = \mathbf{R}_1 \mathbf{s}_1 + \mathbf{R}_m \mathbf{m}$ for a public $\mathbf{u}$, it cannot be invoked on them directly. We therefore isolate a vanishing-constant-coefficient wrapper $\Pi_{\text{many}}^{(\text{ct})}$ (§3.2.1), in the style of [LNP22, §3.2, §4.3], aggregating an arbitrary family of such statements into a single full-ring relation at additive cost $1/|\mathbb{K}|$, a single execution sufficing because our challenges lie in $\mathbb{K} = \mathbb{F}_{q^2}$.
- As a by-product of that aggregation, batching the degree-0 checks with independent per-index challenges avoids the union bound one would otherwise pay, giving soundness error $2/|\mathbb{K}|$ rather than $(K + k)t/|\mathbb{K}|$ (Lemma 3.13). Separating the three batched families with independent challenges, rather than with powers of a single γ , likewise improves the Schwartz–Zippel error of the batching (cf. [NS26, Appendix B.2]).
- We commit to the masking polynomial by applying ABDLOP directly to its coefficients, rather than through a polynomial commitment scheme supporting coefficients in $\mathbb{K}$; our reasons for such approach are discussed in §3.5.
- Handling commitments comes with a cost. We give a modified random-linear-combination reduction which, rather than decomposing the ABDLOP salts alongside the witnesses, resamples them at every decomposition step and certifies the pieces by a batched proof of knowledge; this keeps every accumulator commitment MLWE-hiding, decouples the salt budget from the decomposition depth k , and lowers the bound at which binding must be assumed. Simulatability of the resulting randomized pair, and hence blinding, further requires the rejection sampling of [LNP22, §2.6] recalled in §3.6, whose repetition overhead is a small (≈ 2 – 3) constant factor.
- A subtlety specific to the interactive setting: an aborted rejection-sampling attempt is not discarded but remains in the verifier’s view, and, conditioned on rejection, the mask depends on the very witness it is meant to hide. Since the mask’s Ajtai image is not hiding, we commit to it under ABDLOP and open it only on acceptance, so that each rejected round carries hiding commitments alone while the mask stays bound before the folding challenges are drawn (Remark B.9). The issue disappears under Fiat–Shamir, where aborted attempts are never output, or once we compose the machinery with the statistical hiding Ajtai commitment.
- We give a single consolidated expression for the completeness error (Proposition 4.15) and the knowledge-soundness error of the composed protocol (Proposition 4.17), rather than leaving the various contributions scattered across the individual reductions.
- Composing these pieces, we obtain LatticeBlindFold (Protocol 12), a folding scheme step that is complete, knowledge-sound, and blinding, built entirely from plausible post-quantum (Ring/Module-SIS/LWE) assumptions, and represents the analogue of NovaBlindFold.
- Finally, we isolate as a corollary (Corollary 4.24) an *accumulator-free* variant Π_{LBF}° , obtained by deleting the k blinding pairs the prover samples. These contribute nothing to hiding, which is carried entirely by the Gaussian mask and the blinding block, and are retained in Protocol 12 only so that its input and output types match those of Π_{SuperNeo} and of the folding scheme blueprint of §4.2.1.

Theorem A (LatticeBlindFold, informal: Theorem 4.13, Corollary 4.24). *Let $\mathbb{F}$ be a small prime field of order q , let $\mathbb{K} = \mathbb{F}_{q^2}$, and let $\mathbb{R}_{\mathbb{F}} = \mathbb{F}[X]/(\Phi(X))$ be a cyclotomic ring of degree d . Let $\mathcal{L}$ be the succinct Ajtai commitment that SuperNeo commits witnesses under, let Commit be the ABDLOP commitment scheme, and*

let $\mathcal{C} \subset \mathbb{R}_{\mathbb{F}}$ be a strong sampling set with expansion factor T . Assume MSIS and MLWE are hard over $\mathbb{R}_{\mathbb{F}}$. Then the LatticeBlindFold step $\Pi_{\text{LBF}} = \Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$ (Protocol 12, Definition 4.6) is an interactive reduction

$$\Pi_{\text{LBF}}: \text{CCS}^\circ(b)^K \hookrightarrow \text{CCS}(b, \mathcal{L})^K \longrightarrow \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k,$$

folding K fresh R1CS instance-witness pairs, together with k accumulator instances the prover samples itself, into k committed evaluation claims of norm bound b , such that:

- (1) (Completeness.) It is statistically complete, with error $\varepsilon_{\text{c}}^{\text{LBF}} \leq 9 \cdot 2^{-\lambda}$ uniformly in k and t .
- (2) (Knowledge soundness.) It is a reduction of knowledge, with error

$$\varepsilon_{\text{LBF}} \leq \frac{\mathcal{O}(\log_2 m)}{|\mathbb{K}|} + \frac{W_{\max}(K+k)}{|\mathcal{C}|} + \mathcal{O}((K+k)t) \cdot \varepsilon_{\text{ABDLOP}} + \varepsilon_{\text{bind}},$$

where $\varepsilon_{\text{ABDLOP}}$ collects the knowledge-soundness errors of the ABDLOP proofs of knowledge and $\varepsilon_{\text{bind}}$ the (relaxed) binding errors of $\mathcal{L}$ and Commit. It is negligible in λ whenever MSIS is hard at the relevant bounds and $|\mathbb{K}|, |\mathcal{C}|$ are superpolynomial.

- (3) (Blinding.) If $K = 1$ the reduction is blinding in the sense of Definition 2.11: the verifier's entire view, aborted rejection-sampling attempts included, is simulatable from the public instances alone, with simulator advantage at most $\varepsilon_{\text{LBF-blind}}$.

Moreover, at a matched decomposition parameter k (cf. Remark 4.18), the prover time, verifier time, and communication of Π_{LBF} agree asymptotically with those of Π_{SuperNeo} up to constant factors, the additive overhead being the $n_{\mathbb{F}, \text{in}}$ field elements of the blinding mask's public instance, the public data of the k blinding pairs the prover samples (a further $n_{\mathbb{F}, \text{in}}$ field elements for all k together, plus $\Theta(k)$ commitments, which the $\Theta(k)$ already present in both schemes absorbs into the constants), and the blinded layout of Definition 3.1 (at the worst $n_{\mathbb{F}}$ doubles).

The three clauses are Theorem 4.13, quantified by Propositions 4.15 and 4.17; the efficiency claim is §4.3.3.

Ten restrictions on the construction above are worth flagging immediately, since they qualify the paper's central claim.

- (1) Even though the protocol can be stated and proved to be a reduction of knowledge for general CCS instances, we decide to focus on the R1CS setting (more precisely on those constraint systems arising from circuits) for efficiency reasons; cf. Remark 4.1.
- (2) We prove that Π_{LBF} (Protocol 12) is blinding (Definition 2.11) only when $K = 1$, i.e. exactly one genuine R1CS instance is folded per invocation, alongside k simulated ones, and only for one step (not a full folding scheme). Therefore, we deduce that Π_{LBF} (Protocol 12) introduces zero-knowledge. Folding $K > 1$ genuine instances simultaneously is complete and knowledge-sound but its blinding property is not established here; see Theorem 4.13 and §4.1.1.1.
- (3) Security is proved in the *interactive* setting. Under Fiat–Shamir the $1/|\mathbb{K}|$ -type terms degrade by the random-oracle query bound $Q \approx 2^{80}$, and a quadratic extension no longer suffices at any Q of cryptographic interest. The non-interactive deployment path therefore requires a higher extension degree, $\mathbb{K} = \mathbb{F}_{q^4}$ at $q = 2^{64} - 59$; this is quantified in §4.5, and it is a requirement on the parameters rather than an open problem (the extension degree is, however, the easy half, while the non-interactive figures of that section all assume round-by-round knowledge soundness of the Sum-Check-plus-ABDLOP composition, which we do not prove and which the generic multi-round bound cannot substitute for at $21 + N_{\text{PoK}}$ challenge rounds; cf. item (2) of §5). We instantiate the protocol in two concrete examples, $d = 64$ and $d = 128$, achieving ≈ 116 and ≈ 121 bits of security respectively ($\lambda = 120$, §4.3.4). The two examples must be considered as illustrative, since the blinding error $\varepsilon_{\text{LBF-blind}} \leq 2^{-112}$; cf. Remark 4.23.
- (4) Our efficiency claim is stated at a matched k . Read against Π_{SuperNeo} at its native $k = \Theta(1)$, blinding costs a $\Theta(\log n_{\mathbb{F}})$ factor in prover time, verifier time and communication (Remark 4.18). We regard reducing this to something deployable as the main open engineering problem the paper leaves.
- (5) Our completeness and blinding claims are proved for a single folding step (as this fits our main use-case). Each step is only statistically, not perfectly, complete. The error comes from the attempt-budget caps on *every* rejection-sampling loop the protocol runs, namely the mask loop of Π'_{RLC} (Remark 4.3) together with the loop internal to each ABDLOP proof-of-knowledge invocation, and

is $9 \cdot 2^{-\lambda}$ under the batched default (Proposition 4.15; the counting convention, and the more conservative unbatched figure, are fixed once in §4.1.2.1). We stress that this is an abort probability for the *honest* prover which vanishes under Fiat–Shamir once the attempt caps are lifted (as no verifier is present to enforce them), at the price of an expected- rather than strictly-polynomial-time prover (Remark 2.12). Note that this is a statement about completeness only and that the same caps carry a knowledge-soundness role that Fiat–Shamir does not remove).

- (6) The relaxation of Definition 3.22 compounds across rounds. Each round of the blueprint Protocol 13 $\Pi_{\text{LBF-Fold}}$ contributes a factor $\bar{c}$ from Lemma 3.6(2) and a factor $(\rho_i - \rho'_i)$ from the coordinate-wise inversion of Lemma 2.19, so the number of sequential rounds the extraction supports is $\mathcal{O}(1)$, and small (Remark 4.8). Once again, a full treatment of *recursive* folding, i.e. n sequential rounds as in an IVC scheme, is out of scope of this paper, but we wish to mention that this is a limitation for obtaining such a folding scheme, which is not merely confined to the proof technique: an IVC use of $\Pi_{\text{LBF-Fold}}$ requires the exact ℓ_2 -norm certificate on the fresh salts sketched there, which we do not instantiate.
- (7) We consider, throughout the paper, a prime $q \equiv 5 \pmod{8}$ and consider a cyclotomic ring of the form $\mathbb{Z}[X]/(x^d + 1)$ for d a power of 2; we believe that most of the discussions can be generalized to a general cyclotomic ring, but we do not discuss them here.
- (8) Every R1CS structure and witness below follows the blinded layout of Definition 3.1, so that the input instance component $c_1 = \mathcal{L}(\mathbf{z}_1)$ is hiding (Lemma 3.3). This is what upgrades the blinding property of Theorem 4.13 to a zero-knowledge statement, but the layout does not do it alone: the upgrade is stated over the unpadded relation of Definition 3.28, the padding being applied by the prover as the embedding of Equation (3.19). Stated instead over the committed, already-padded relation, the layout buys nothing, since the adversary then supplies the blinding block itself and may set it to zero (Remark 3.27).
- (9) What Π_{LBF} produces is k instances of $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)$ together with the $N_{\text{PoK}}^{(1)} \approx 10^2$ ABDLOP openings that certify them, some 5–8 MB at the parameters of §4.3.4. Someone must still check those claims. Two routes exist, and we instantiate neither in this work: either the downstream verifier consumes the openings as they are, at that communication cost, or the ABDLOP verification equation is arithmetized so that it can itself be folded or proved succinctly. Both are outside the scope of this paper; cf. item (7) of §5.
- (10) Relatedly, Π_{LBF} is not a folding scheme in the usual instance-in/instance-out sense. Its input relation is the *uncommitted*, unpadded $\text{CCS}^\circ(b)$ of Definition 3.28, so c_1 is a prover message rather than an instance component (Remark 3.27); this is exactly what makes the zero-knowledge reading available, and exactly what breaks the type match with a folding step. Protocol 13 is a blueprint, its recursive branch is not proved, and by Remark 4.8 the extraction survives only $\mathcal{O}(1)$ rounds in any case. The object we prove things about is a one-shot wrapper for a finished circuit, which is the use case of §1.2.1.

1.4. Paper organization. The paper is organized as follows. §2 recalls the interactive-reduction framework of SuperNeo, the notion of a blinding folding reduction, strong sampling sets, and the two commitment schemes we rely on (the succinct Ajtai scheme and the hiding ABDLOP scheme [BDL⁺16, LNP22]), and generalizes the interactive-reduction formalism to allow completeness errors. §3 develops the components that make SuperNeo’s Sum-Check zero-knowledge: Libra-style masking, the componentwise instantiation of ABDLOP over $\mathbb{R}_{\mathbb{K}}$, the companion procedures from the ABDLOP proofs of knowledge, and the rejection sampling recalled in §3.6. §4 assembles the three modified reductions Π'_{R1CS} , Π'_{RLC} and Π'_{DEC} into the LatticeBlindFold protocol and proves it complete, knowledge-sound, and blinding, with consolidated error expressions in Propositions 4.15 and 4.17. §4.3.3 collects the complexity counts, and §4.4 addresses a Corollary 4.24, an *accumulator-free* variant of Π_{LBF} . §5 discusses open questions and follow-up work. Deferred proofs are collected in §B.

2. RECALLS AND PRELIMINARIES

We collect in this section recalls from SuperNeo [NS26] as well as all the necessary technical background.

2.1. Notation and Global Parameters. We consolidate here the notation and conventions used throughout this work, expanding upon the baseline established in [NS26].

- *Asymptotics and sets.* λ denotes the security parameter, fixed to $\lambda = 120$ throughout. We write $\text{negl}(\lambda)$ and $\text{poly}(\lambda)$ for a negligible and a polynomial function of λ , PPT and EPT for probabilistic polynomial time and expected probabilistic polynomial time, and $[n] := \{1, \dots, n\}$. All asymptotics implicitly depend on λ .
- *Polynomials and evaluations.* $\mathbb{F}^d[X_1, \dots, X_n]$ is the ring of multivariate polynomials over $\mathbb{F}$ in n variables with individual degree bound d per variable, the superscript being omitted when the degree is unconstrained, and $ZS_\ell \subseteq \mathbb{F}[X_1, \dots, X_\ell]$ is the set of polynomials vanishing on the Boolean hypercube. We write $\text{eq}(x, y) := \prod_{i=1}^\ell (x_i y_i + (1 - x_i)(1 - y_i))$ for the equality polynomial and, for $v \in \mathbb{F}^n$, $\tilde{v}(X) = \sum_{j \in \{0,1\}^{\log n}} \text{eq}(X, j) \cdot v_j$ for the unique multilinear extension of v . For the Sum-Check polynomial $Q(\vec{X})$ we write $D_{\max} := \max\{u + 1, 2b, 2\}$ for its maximum per-variable degree, where u is the CCS degree and b the norm bound fixed below; the $\text{eq}(\vec{X}, \alpha)$ factor contributes one extra per-variable degree to *both* summands it multiplies, whence $u + 1$ on the F term and $2b$ on the NC term of Protocol 6. At our parameters $u = b = 2$, so $D_{\max} = 4$. Finally, $\vec{\gamma}^{(1)} \in \mathbb{K}^{K+k}$, $\vec{\gamma}^{(2)} \in \mathbb{K}^t$ and $\vec{\gamma}^{(3)} \in \mathbb{K}^d$ are the three independent *tensoring batching challenge* vectors of Protocol 6, indexed respectively by instance, by structural matrix and by coefficient slot, the weight attached to the triple (i, j, ℓ) being $\gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)}$ and that attached to a single instance index i being $\gamma_i^{(1)}$. Since [NS26, Appendix B.2] uses instead the powers of a single challenge, the two differ only in the Schwartz–Zippel degree they incur, and Lemma 4.16 is where that difference is paid for.
- *Rings, fields and dimensions.* $\mathbb{F}$ is a finite field of prime order q and $\mathbb{K}$ a low-degree extension of $\mathbb{F}$ with $1/|\mathbb{K}| = \text{negl}(\lambda)$, so that $\mathbb{F}$ is naturally a subfield of $\mathbb{K}$. We let $\Phi(X) := X^d + \Phi_{d-1}X^{d-1} + \dots + \Phi_0 \in \mathbb{F}[X]$ be the η -th cyclotomic polynomial, of degree d , and set $\mathbb{R}_{\mathbb{F}} := \mathbb{F}[X]/(\Phi(X))$ and $\mathbb{R}_{\mathbb{K}} := \mathbb{K}[X]/(\Phi(X))$, the former embedding naturally into the latter; we fix the ring automorphism $\sigma: X \mapsto X^{-1}$; in this note we focus on the case $\Phi(X) = X^d + 1$, i.e. $\eta = 2d$ and d is a power of 2. We write m for the number of constraints, $n_{\mathbb{R}}$ for the length of a ring vector and $n_{\mathbb{F}} = d \cdot n_{\mathbb{R}}$ for the corresponding field vector length, with $n_{\mathbb{R}, \text{in}}$ and $n_{\mathbb{F}, \text{in}}$ the respective input lengths; u denotes a polynomial degree bound, t the number of structural matrices, $k, K \in \mathbb{Z}_{\geq 1}$ the numbers of instances to be folded, and $b, B \in \mathbb{Z}_{\geq 2}$, with $B = b^k < q/2$, the norm bounds. We write $n_{\mathbb{F}, \text{bl}}$ for the length of the *blinding block*, the trailing segment of the witness whose sole role is to randomize the Ajtai image $\mathcal{L}(\mathbf{z})$ (Definition 3.1); we take $d \mid n_{\mathbb{F}, \text{bl}}$ and set $n_{\mathbb{R}, \text{bl}} := n_{\mathbb{F}, \text{bl}}/d$, so that the block occupies whole ring elements and the corresponding column block $A_{\text{bl}} \in \mathbb{R}_{\mathbb{F}}^{\kappa \times n_{\mathbb{R}, \text{bl}}}$ of the commitment matrix is well defined. Both $n_{\mathbb{F}, \text{in}}$ and $n_{\mathbb{F}, \text{bl}}$ are counted inside $n_{\mathbb{F}}$.
- *Commitments and salts.* κ and κ' denote the module ranks of the Ajtai and of the BDLOP components of the ABDLOP commitment scheme (§2.5), and $\kappa_{\text{com}} := \kappa + \kappa'$ the total number of ring components of an ABDLOP commitment $C_j \in \mathbb{R}_{\mathbb{F}}^{\kappa_{\text{com}}}$ (cf. §2.5.2). The BDLOP rank is by definition the number of message slots the commitment carries, so $\kappa' = \ell$, and we use the two interchangeably. Commitment salts are always written $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2) \in \mathbb{R}_{\mathbb{F}}^{m_1} \times \mathbb{R}_{\mathbb{F}}^{m_2}$, the subscript $l \in [2]$ selecting the Ajtai or the BDLOP block. Salts and commitments are always $\mathbb{R}_{\mathbb{F}}$ -objects and *never* $\mathbb{R}_{\mathbb{K}}$ -ones: by §3.3 an $\mathbb{R}_{\mathbb{K}}$ -message is committed as the pair (a, b) of its $\{1, Y\}$ -coordinates, occupying two message slots of a single $\mathbb{R}_{\mathbb{F}}$ -commitment, and nothing in this paper requires MLWE or MSIS over $\mathbb{R}_{\mathbb{K}}$. Every ABDLOP commitment issued here carries its message in the *BDLOP* slot, i.e. $\ell^{(n)} \geq 1$ and $\mathbf{t}_B^{(n)} = \mathbf{B}\mathbf{s}_2^{(n)} + \mathbf{m}^{(n)}$, never in the Ajtai slot $\mathbf{s}_1$; both $\mathbf{s}_1$ and $\mathbf{s}_2$ are salts. We stress this because [LNP22]’s own instantiations (loc. cit., Figs. 11, 13, 15) take $\ell = 0$ and place the message in the Ajtai part instead. We denote the challenge coefficient bound in the ABDLOP proofs of knowledge (cf. [LNP22]) with β_{ch} . Finally, the triple (ℓ, m_1, m_2) so fixed governs *every* ABDLOP commitment of this paper, the mask-block commitment $C_{\mathbf{y}, 0}$ of Protocol 7 included: its message is too long for a single commitment, so it is carried by a tuple of $\tilde{N}$ commitments at these same parameters rather than by one wider commitment (Remark 4.21).
- *Coefficients, decompositions and norms.* The coefficient vector of $a \in \mathbb{R}_{\mathbb{F}}$ is $\text{cf}(a) \in \mathbb{F}^d$ and, for $z \in \mathbb{R}_{\mathbb{F}}^m$, $\text{cf}(z) \in \mathbb{F}^{d \times m}$ is the matrix of the coefficient vectors, whose ℓ -th row is $\text{cf}(z)_\ell$; the constant term is $\text{ct}(a) := \text{cf}(a)_1 \in \mathbb{F}$, extended component-wise to vectors. Both maps are defined identically over $\mathbb{R}_{\mathbb{K}}$. For $a \in \mathbb{F}$, $\|a\|_\infty$ is taken from the symmetric representative in $[-(q-1)/2, (q-1)/2]$; for a

vector it is the maximum over the components, and for elements of $R_{\mathbb{F}}$ or $R_{\mathbb{F}}^m$ it applies component-wise to the field coefficient representation. Finally, $\text{split}_b: \mathbb{F}^m \rightarrow (\mathbb{F}^m)^k$ is the b -ary decomposition: for $\|z\|_{\infty} < b^k$ it returns $(z_1, \dots, z_k)$ with $z = \sum_{i=1}^k b^{i-1} z_i$ and $\|z_i\|_{\infty} < b$.

- *Rejection sampling.* M is the repetition rate of the rejection sampling function Rej_1 (Lemma 3.20) that we use to achieve blinding, and $W_{\max} := \lceil \lambda / \log_2(M/(M-1)) \rceil$ the *attempt budget*, the publicly fixed maximum number of iterations of the Rej_1 loop of Protocol 7. We write N_{PoK} for the total number of ABDLOP proof-of-knowledge invocations made by one execution of Π_{LBF} , counted in Proposition 4.15; it is the quantity that multiplies the per-invocation completeness error.

Notation 2.1. *A warning on overloaded symbols, since several are inherited from the two sources we build on and cannot be renamed without breaking the correspondence with them. The letter ℓ is used for the number of Sum-Check variables $\log_2(m)$, for the BDLOP message length of §2.5.2, for the coefficient index in $\text{cf}(\cdot)_{\ell}$, and for the challenge-vector dimension of Lemma 2.19; l , separately, is the number of irreducible factors of $X^d + 1$ modulo q . The letter $\mathbf{s}$ denotes a structure, $\mathbf{s}$ a Gaussian width, and $\mathbf{s}_1, \mathbf{s}_2$ the commitment salts. The letter m is the number of constraints, and in the ABDLOP literature also the committed message, which is why Lemma 2.31 renames the latter to y . Finally κ is the Ajtai rank here and the challenge coefficient bound in [LNP22]. In each case the intended meaning is fixed by the surrounding subsection, and we flag the ambiguity at the point of use where it could plausibly bite.*

The parameters above, together with the objects fixed below, are what protocol statements later refer to as the global parameters.

Definition 2.2 (Global parameters). *In addition to $\mathbb{F}, \mathbb{K}, d, R_{\mathbb{F}}, R_{\mathbb{K}}, m, n_{\mathbb{F}}, n_{\mathbb{R}}, n_{\mathbb{F}, \text{in}}, n_{\mathbb{R}, \text{in}}, u, t, k, K, b, B$ fixed above, we also fix:*

- A structure $\mathbf{s}$, formally defined in §2.4.1 below as a collection of structural matrices together with a degree- u polynomial.
- A strong sampling set $\mathcal{C} \subseteq R_{\mathbb{F}}$ (Definition 2.13 below) with expansion factor T , such that $(K + k)T(b - 1) < B$ and $1/|\mathcal{C}| = \text{negl}(\lambda)$.
- A ring commitment scheme $\text{com} := (\text{Setup}, \text{Commit})$ (Definition 2.27 below), homomorphic and $(2B, \mathcal{C})$ -relaxed binding. For $\text{pp} \leftarrow \text{Setup}(1^{\lambda}, m)$, we write $\mathcal{L} := \text{Commit}(\text{pp}, \cdot) : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow \mathbb{C}$ for the induced $R_{\mathbb{F}}$ -module homomorphism. We additionally require $\mathcal{L}$ to be hiding on the blinding block in the sense of Lemma 3.3. (This is a condition on the witness layout of Definition 3.1 and on $n_{\mathbb{R}, \text{bl}}$, not on the commitment scheme, which remains the plain compact Ajtai scheme of §2.5.1.)
- The trivial $R_{\mathbb{F}}$ -module homomorphism $\mathcal{L}_{\text{in}} : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow R_{\mathbb{F}}^{n_{\mathbb{R}, \text{in}}}$, projecting onto the first $n_{\mathbb{R}, \text{in}}$ columns.

2.2. Interactive reductions. We recall here the concept of interactive reductions, which generalize traditional reductions to an interactive setting between a prover and a verifier. The following definitions adapt the framework introduced in [NS26, §4.6] to our setting and extend the notion of blinding to interactive reductions.

Definition 2.3 (Interactive Reductions, [NS26, Definition 9]). *An interactive reduction from relation $\mathcal{R}_1$ to relation $\mathcal{R}_2$ (over parameters, structure, instance, and witness tuples) is defined by a tuple of PPT algorithms $(\mathcal{G}, \mathcal{K}, \mathcal{P}, \mathcal{V})$:*

- $\mathcal{G}(1^{\lambda}, \text{sz}) \rightarrow \text{pp}$: Generates public parameters pp given a security parameter λ and size parameters sz .
- $\mathcal{K}(\text{pp}, \mathbf{s}) \rightarrow (\text{pk}, \text{vk})$: Deterministically outputs a prover key pk and verifier key vk derived from pp and a given structure $\mathbf{s}$.
- $\mathcal{P}(\text{pk}, u_1, w_1) \rightarrow (u_2, w_2)$: Interactively reduces the check of $(\text{pp}, \mathbf{s}, u_1, w_1) \in \mathcal{R}_1$ into a new claim $(\text{pp}, \mathbf{s}, u_2, w_2) \in \mathcal{R}_2$.
- $\mathcal{V}(\text{vk}, u_1) \rightarrow u_2$: Interactively reduces instance $u_1 \in \mathcal{R}_1$ into a new instance $u_2 \in \mathcal{R}_2$.

The complete execution of the protocol is denoted $\langle \mathcal{P}, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, w_1)$, which ultimately outputs the verifier's new instance u_2 and the prover's corresponding witness w_2 ; we treat $\langle \mathcal{P}, \mathcal{V} \rangle$ as a function that takes as input $((\text{pk}, \text{vk}), u_1, w_1)$ and runs the interaction on the prover's input (pk, u_1, w_1) and the verifier's input (vk, u_1) .

Definition 2.4 (Reduction of Knowledge (RoK), [NS26, Definition 9]). *An interactive reduction $(\mathcal{G}, \mathcal{K}, \mathcal{P}, \mathcal{V})$ is a Reduction of Knowledge (RoK) if it achieves:*

- (1) **Completeness (with error ε_c)**: For any EPT adversary $\mathcal{A}$, given $\text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz})$, $(\text{s}, u_1, w_1) \leftarrow \mathcal{A}(\text{pp})$ such that $(\text{pp}, \text{s}, u_1, w_1) \in \mathcal{R}_1$, we have that, except with probability at most $\varepsilon_c = \varepsilon_c(\lambda)$ over the honest parties' coins, the prover's output instance is equal to the verifier's output instance u_2 , and

$$(\text{pp}, \text{s}, \langle \mathcal{P}, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, w_1)) \in \mathcal{R}_2.$$

We call the reduction perfectly complete when $\varepsilon_c = 0$ and statistically complete when $\varepsilon_c = \text{negl}(\lambda)$, and we say that it is ε_c -complete to record the error explicitly.

- (2) **Knowledge soundness**: For any EPT adversary $(\mathcal{A}, \mathcal{P}^*)$, there exists an EPT extractor $\mathcal{E}$ such that if the success probability of the adversary

$$\epsilon(\mathcal{A}, \mathcal{P}^*) := \Pr \left[(\text{pp}, \text{s}, \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st})) \in \mathcal{R}_2 \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \end{array} \right] \geq 1/\text{poly}(\lambda),$$

then we have that

$$\Pr \left[(\text{pp}, \text{s}, u_1, w_1) \in \mathcal{R}_1 \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ w_1 \leftarrow \mathcal{E}(\text{pp}, \text{s}, u_1, \text{st}) \end{array} \right] \geq \epsilon(\mathcal{A}, \mathcal{P}^*) - \text{negl}(\lambda).$$

- (3) **Public Coin**: All of the verifier's messages are uniformly random strings of some prescribed length. Furthermore, the verifier's messages contain all of the random coins (randomness) used by the verifier.

Remark 2.5. SuperNeo's reductions are perfectly complete, and [NS26, Definition 9] accordingly states completeness without an error term. Ours cannot be, by construction. Let us notice that completeness errors accumulate additively under composition (cf. Proposition 2.8).

We make use of two conventions:

- (1) The loops are capped. An uncapped loop is complete with error 0 but only expected-polynomial-time, which breaks the extraction arguments (Remark B.2) and, in the interactive setting, is not enforceable by $\mathcal{V}$. We cap every loop at $W_{\max}$ for Protocol 7 and charge the resulting exhaustion probability to ε_c . The two errors are of the same shape, $(1 - 1/M)^W \leq 2^{-\lambda}$, but arise from unrelated loops and must be counted separately.
- (2) The failure event is public and witness-independent. Budget exhaustion is an abort declared by the honest prover; it is not a verifier rejection of a valid statement, and, by Lemma 3.20, its probability is (up to statistical distance $2^{-\lambda}$) independent of the witness being folded. This is what lets the completeness error stay outside the blinding analysis: a simulator reproduces the abort pattern by tossing coins with the same, publicly known, bias, without knowing the witness (cf. Remark 2.12).

To build composite reductions, we rely on the notions of *weak* and *strong* reductions. We fix relations $\mathcal{R}_1, \mathcal{R}'_1, \mathcal{R}_2$, and $\mathcal{R}'_2$ over public parameters, structure, instance, and witness tuples such that $\mathcal{R}_1 \subseteq \mathcal{R}'_1$ and $\mathcal{R}_2 \subseteq \mathcal{R}'_2$. Let $\mathcal{U}_i$ and $\mathcal{W}_i$ be, respectively, the ambient instance and witness spaces of $\mathcal{R}_i$, $i \in [2]$.

Both notions below are stated relative to a pair of functions: an *instance* projection ϕ , as in [NS26, Definitions 16–17], and a *witness* projection ψ , which is new here. In loc. cit. the two agreement conditions are stated as equality of the extracted witnesses, which is legitimate there because a CE witness is the vector $\mathbf{z}$ alone, and $\mathbf{z}$ is determined by $\phi(u) = (c_i)_i$ through the (relaxed) binding of $\mathcal{L}$. Our witnesses carry two further components, the evaluation hints $\{y_{i,j}\}$ and the commitment salts $\{s_{i,j}\}$, and neither is a function of $\phi(u)$. The hints depend on the Sum-Check point r' , which two runs of Π'_{R1CS} do not share, and the salts are resampled afresh at every commitment. Verbatim witness equality therefore fails for essentially every pair of runs, and with it the second clause of Definition 2.6 and the hypothesis of Definition 2.7.(2). We accordingly ask for agreement only under ψ , which is all the extraction arguments consume; taking $\psi = \text{id}$ recovers [NS26, Definitions 16–17] exactly.

Definition 2.6 (Weak Interactive Reductions; cf. [NS26, Definition 16]). An interactive reduction $\Pi : \mathcal{R}_1 \rightarrow \mathcal{R}_2$, defined by PPT algorithms $(\mathcal{G}, \mathcal{K}, \mathcal{P}, \mathcal{V})$, is *weak* (with completeness error ε_c) if it is ε_c -complete, public coin, and there exist functions $\phi : \mathcal{U}_1 \rightarrow \mathbb{C}$ and $\psi : \mathcal{W}_1 \rightarrow \mathbb{D}$ (for arbitrary spaces $\mathbb{C}, \mathbb{D}$) such that for any

EPT adversary $(\mathcal{A}, \mathcal{P}^*)$, there exists an EPT extractor $\mathcal{E}$ such that if the success probability of the adversary $\epsilon(\mathcal{A}, \mathcal{P}^*) \geq 1/\text{poly}(\lambda)$, then

$$\Pr \left[(\text{pp}, \text{s}, u_1, w_1) \in \mathcal{R}'_1 \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ w_1 \leftarrow \mathcal{E}(\text{pp}, \text{s}, u_1, \text{st}) \end{array} \right] \geq \epsilon(\mathcal{A}, \mathcal{P}^*) - \text{negl}(\lambda).$$

and if $\mathcal{A} := (\mathcal{B}, \mathcal{B}')$ such that

$$\Pr \left[\begin{array}{l} u_1, u'_1 \neq \perp \\ \Downarrow \\ \phi(u_1) = \phi(u'_1) \end{array} \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, \text{st}^*) \leftarrow \mathcal{B}(\text{pp}) \\ (u_1, \text{st}) \leftarrow \mathcal{B}'(\text{st}^*) \\ (u'_1, \text{st}') \leftarrow \mathcal{B}'(\text{st}^*) \end{array} \right] = 1,$$

then

$$\Pr \left[\begin{array}{l} w_1, w'_1 \neq \perp \\ \wedge \psi(w_1) \neq \psi(w'_1) \end{array} \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, \text{st}^*) \leftarrow \mathcal{B}(\text{pp}) \\ (u_1, \text{st}) \leftarrow \mathcal{B}'(\text{st}^*) \\ w_1 \leftarrow \mathcal{E}(\text{pp}, \text{s}, u_1, \text{st}) \\ \\ (u'_1, \text{st}') \leftarrow \mathcal{B}'(\text{st}^*) \\ w'_1 \leftarrow \mathcal{E}(\text{pp}, \text{s}, u'_1, \text{st}') \end{array} \right] \leq \text{negl}(\lambda)$$

Definition 2.7 (Strong Interactive Reductions, [NS26, Definition 17]). An interactive reduction $\Pi : \mathcal{R}_1 \rightarrow \mathcal{R}_2$, defined by PPT algorithms $(\mathcal{G}, \mathcal{K}, \mathcal{P}, \mathcal{V})$, is strong (with completeness error ϵ_c) if it is ϵ_c -complete, public coin, and there exist functions $\phi : \mathcal{U}_2 \rightarrow \mathbb{C}$ and $\psi : \mathcal{W}_2 \rightarrow \mathbb{D}$ (for arbitrary spaces $\mathbb{C}, \mathbb{D}$) such that

(1) For any EPT adversary $(\mathcal{A}, \mathcal{P}^*)$,

$$\Pr \left[\begin{array}{l} u_2, u'_2 \neq \perp \\ \Downarrow \\ \phi(u_2) = \phi(u'_2) \end{array} \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}_1) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ (u_2, w_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}) \\ (u'_2, w'_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}) \end{array} \right] = 1$$

(2) For any EPT adversary $(\mathcal{A}, \mathcal{P}^*)$, there exists an EPT extractor $\mathcal{E}$ such that if

$$\epsilon'(\mathcal{A}, \mathcal{P}^*) := \Pr \left[(\text{pp}, \text{s}, \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st})) \in \mathcal{R}'_2 \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \end{array} \right] \geq 1/\text{poly}(\lambda),$$

and

$$\Pr \left[\begin{array}{l} w_2, w'_2 \neq \perp \\ \wedge \\ \psi(w_2) \neq \psi(w'_2) \end{array} \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ (u_2, w_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}) \\ (u'_2, w'_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}) \end{array} \right] \leq \text{negl}(\lambda)$$

then we have that

$$\Pr \left[(\text{pp}, \text{s}, u_1, w_1) \in \mathcal{R}_1 \mid \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ w_1 \leftarrow \mathcal{E}(\text{pp}, \text{s}, u_1, \text{st}) \end{array} \right] \geq \epsilon'(\mathcal{A}, \mathcal{P}^*) - \text{negl}(\lambda).$$

Proposition 2.8 (Strong-Weak Composition, [NS26, Theorem 12]). Consider relations $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}'_2$ and $\mathcal{R}_3$ over public parameters, structure, instance, and witness tuples such that $\mathcal{R}_2 \subseteq \mathcal{R}'_2$. Let $\mathcal{U}_2$ and $\mathcal{W}_2$ be the ambient instance and witness spaces of $\mathcal{R}_2$. Consider interactive reductions $\Pi_1 : \mathcal{R}_1 \rightarrow \mathcal{R}_2$ ($\mathcal{R}'_2$), $\Pi_2 : \mathcal{R}_2$ ($\mathcal{R}'_2$) $\rightarrow \mathcal{R}_3$ such that

- (1) Π_1 is strong with respect to a pair of functions $\phi : \mathcal{U}_2 \rightarrow \mathbb{C}$, $\psi : \mathcal{W}_2 \rightarrow \mathbb{D}$ and
- (2) Π_2 is weak with respect to the same pair (ϕ, ψ) ,

then the sequential composition $\Pi_2 \circ \Pi_1 : \mathcal{R}_1 \rightarrow \mathcal{R}_3$ is a reduction of knowledge. Moreover, if Π_1 and Π_2 have completeness errors $\varepsilon_c^{(1)}$ and $\varepsilon_c^{(2)}$ respectively, then $\Pi_2 \circ \Pi_1$ has completeness error at most $\varepsilon_c^{(1)} + \varepsilon_c^{(2)}$.

The completeness clause is not part of [NS26, Theorem 12], whose reductions are perfectly complete, but it requires no new argument. The composed honest execution runs Π_1 and then Π_2 on Π_1 's output, so it fails only if one of the two does, and a union bound applies. We record the clause explicitly because, unlike in loc. cit., in our setting both summands are nonzero. The ψ -clause requires no change to the argument of [NS26, Theorem 12] either. That proof uses witness agreement in exactly one place, namely to discharge the hypothesis of Definition 2.7.(2) for Π_1 from the conclusion of Definition 2.6 for Π_2 ; since both are now stated under the same ψ , the implication chains verbatim. It is for this reason that Π_1 and Π_2 must agree on ψ and not merely on ϕ .

Proposition 2.9 (Sequential composition of reductions of knowledge, [KP22, Theorem 5]). *If $\Pi_1 : \mathcal{R}_1 \rightarrow \mathcal{R}_2$ and $\Pi_2 : \mathcal{R}_2 \rightarrow \mathcal{R}_3$ are reductions of knowledge with knowledge-soundness errors $\varepsilon_1, \varepsilon_2$ and completeness errors $\varepsilon_c^{(1)}, \varepsilon_c^{(2)}$, then $\Pi_2 \circ \Pi_1 : \mathcal{R}_1 \rightarrow \mathcal{R}_3$ is a reduction of knowledge with errors $\varepsilon_1 + \varepsilon_2$ and $\varepsilon_c^{(1)} + \varepsilon_c^{(2)}$.*

The completeness clause is again not in loc. cit., whose reductions are perfectly complete, and is the same union bound recorded above. We use Proposition 2.8 for the inner composition $\Pi'_{\text{RLC}} \circ \Pi'_{\text{RCS}}$, which is genuinely weak-after-strong, and Proposition 2.9 for the outer one with Π'_{DEC} , both factors being reductions of knowledge outright.

Definition 2.10 (Folding scheme). *A folding scheme for a relation $\mathcal{R}$ is defined as a reduction of knowledge mapping $\mathcal{R} \times \mathcal{R}_{\text{ACC}} \rightarrow \mathcal{R}_{\text{ACC}}$ for an accumulator relation $\mathcal{R}_{\text{ACC}}$.*

We write now the definition of blinding interactive reduction. By abuse of notation, we will refer to an interactive reduction of the form $\Pi_{\text{Fold}} : \mathcal{R}_1 \rightarrow \mathcal{R}_2$, for $\mathcal{R}_1 = \mathcal{R}^l \times \mathcal{R}'$ (for any $l \geq 1$), as a folding interactive relation.

Definition 2.11 (Blinding Folding Interactive Reduction). *A folding interactive reduction $\Pi : \mathcal{R}_1 = \mathcal{R}^l \times \mathcal{R}' \rightarrow \mathcal{R}_2$, more precisely $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{P}, \mathcal{V})$ for $(\mathcal{R}_1, \mathcal{R}_2, \mathbf{s}, \lambda)$, is blinding if there exists a sampling algorithm $\text{sample}_{\mathcal{R}}$ for instance-witness pairs in $\mathcal{R}$ such that for any honest-but-curious verifier $\mathcal{V}^*$ (modeled as a deterministic algorithm $\mathcal{V}^*(\text{transcript}; \omega)$ taking as auxiliary input a coin string $\omega \xleftarrow{\$} \Omega$ sampled uniformly at random and independently of $\mathcal{P}$, from which every protocol challenge is deterministically derived, by a fixed public map whose pushforward of the uniform distribution on Ω is the honest challenge distribution of Π , so that $\mathcal{V}^*$ may compute its challenges early but not bias them), there exists an EPT simulator $\mathcal{S}$ for which the following holds.*

For any PPT adversary $\mathcal{A}$ given

- $\text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz})$ and $((u_1, w_1), (u_2, w_2)) \leftarrow \mathcal{A}(\text{pp})$ such that $(\text{pp}, \mathbf{s}, u_1, w_1) \in \mathcal{R}^{l-1}$, and $(\text{pp}, \mathbf{s}, u_2, w_2) \in \mathcal{R}'$, and
- for $(\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \mathbf{s})$,

we have

$$\left\{ (\text{st}, w) \left| \begin{array}{l} (u_r, w_r) \xleftarrow{\$} \text{sample}_{\mathcal{R}}(\text{pp}, \mathbf{s}), \\ (\text{st}, (u, w)) \leftarrow \langle \mathcal{P}, \mathcal{V}^* \rangle((\text{pk}, \text{vk}), ((u_r, u_1), u_2), ((w_r, w_1), w_2)) \end{array} \right. \right\} \cong \{ (\text{st}, w) \mid (\text{st}, w) \leftarrow \mathcal{S}(\text{pp}, \mathbf{s}, (u_1, u_2)) \},$$

where st denotes the (arbitrary) output of $\mathcal{V}^*$.

As in [KS23, Definition 14], $\mathcal{V}^*$ is deterministic given its randomness, exactly as in the public-coin setting of §2; this is what lets the simulator, which does not have oracle access to $\mathcal{V}^*$'s internal coins, reproduce an identically-distributed view by sampling fresh coins of its own and running the same deterministic map on them. Clearly, for a (multi-)folding scheme, we recover the definition of blinding in [KS23, Definition 14].

In the setting of this paper, $l = 1$, $\mathcal{R}' = \text{CCS}(b, \mathcal{L})$ (Definition 2.21), and $\mathcal{R} = \mathcal{R}_2 = \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$, (Definitions 3.21).

Remark 2.12 (Aborts and blinding). *Since $\langle \mathcal{P}, \mathcal{V}^* \rangle$ may now abort (Remark 2.5), the left-hand distribution of Definition 2.11 is over all executions, aborting ones included, and $\mathcal{S}$ must reproduce it as such rather than only its non-aborting part. This costs nothing, for two reasons*

- First, the abort pattern is simulatable on its own. Whether a given attempt of any of our rejection loops accepts is, by Lemma 3.20 (resp. its ABDLOP analogue), a coin of publicly known bias $1/M$ (resp. $1/M_{\text{PoK}}$) that is within statistical distance $2^{-\lambda}$ of being independent of the secret. $\mathcal{S}$ tosses its own coins with that bias, and aborts where they say to.
- Second, an aborted attempt is not discarded from $\mathcal{V}^*$'s view in the interactive setting. It leaves behind whatever the prover already sent in that attempt. This is exactly why the mask's Ajtai image c_y is committed under ABDLOP at Step 6 of Protocol 7 and opened only on acceptance (Remark B.9), and why the ABDLOP proofs retransmit only a fresh $(\mathbf{w}, \mathbf{v})$ on a retry, never a fresh commitment. A rejected round therefore carries hiding commitments and nothing else, and the hybrid argument of Theorem 4.13 applies to it verbatim.

In the end, under Fiat–Shamir the second point is no more meaningful (at the level of completeness), as aborted attempts are simply never output. We stress that this removes the blinding cost of aborts, not the completeness error itself. The latter disappears only if the attempt caps are lifted as well, in which case the honest prover becomes expected polynomial time rather than strictly PPT. That is harmless non-interactively, as there is no verifier to enforce a cap, and the prover simply hashes again, but it is a change of prover model, not a free consequence of Fiat–Shamir; cf. Remark B.2.

2.3. Strong Sampling Sets: Controlling Norm Growth in Folding. In folding schemes, provers and verifiers linearly combine instances using random coefficients ρ_i , such that $z_{\text{out}} = \sum_i \rho_i z_i$. If these challenges are sampled from the entire ring $\mathbb{R}_{\mathbb{F}}$, the resulting norms can explode exponentially, compromising the underlying MSIS hardness. To prevent this, we restrict the challenges to a *strong sampling set*.

Definition 2.13 ((Strong sampling sets, [NS26, Definition 6])). *A subset $\mathcal{C} \subseteq \mathbb{R}_{\mathbb{F}}$ is a strong sampling set if it satisfies two conditions:*

- (1) *For any distinct elements $a, b \in \mathcal{C}$, the difference $a - b$ has an infinity norm strictly less than the low-norm invertibility threshold $\mathbf{b}_{\text{inv}}$ (Proposition 2.14), guaranteeing that $a - b$ is invertible over $\mathbb{R}_{\mathbb{F}}$.*
- (2) *Its expansion factor, defined as $T(\mathcal{C}) := \max_{v \in \mathbb{R}_{\mathbb{F}}, \rho \in \mathcal{C}} \frac{\|\rho v\|_{\infty}}{\|v\|_{\infty}}$, is bounded by $\mathcal{O}(\phi(\eta) \cdot \max_{\rho \in \mathcal{C}} \|\rho\|_{\infty})$.*

Proposition 2.14 (Low norm invertibility [LS17, Theorem 1.1, Conjecture 2.6]). *Let z divide η , with $q \equiv 1 \pmod{z}$ and $\text{ord}_{\eta}(q) = \eta/z$. Define $\mathbf{b}_{\text{inv}} := 1/\sqrt{\tau(z)} \cdot q^{1/\phi(z)}$ (where $\tau(z) = z$ for odd z , or $z/2$ for even z). Any non-zero element $a \in \mathbb{R}_{\mathbb{F}}$ satisfying $\|a\|_{\infty} < \mathbf{b}_{\text{inv}}$ is guaranteed to be invertible.*

Remark 2.15. *Note that, since Definition 2.13 bounds $\|a - b\|_{\infty} < \mathbf{b}_{\text{inv}}$ for distinct $a, b \in \mathcal{C}$, and since in our instantiations every $\rho \in \mathcal{C}$ itself satisfies $\|\rho\|_{\infty} < \mathbf{b}_{\text{inv}}$, (as $0 \in \mathcal{C}$ or not, this is the same small-coefficient bound used to construct $\mathcal{C}$), Proposition 2.14 applies directly to any single $\rho \in \mathcal{C} - \{0\}$, that is, every nonzero element of $\mathcal{C}$ is invertible in $\mathbb{R}_{\mathbb{F}}$.*

By establishing an expansion factor, we ensure that multiplying any vector v by a challenge $\rho \in \mathcal{C}$ safely bounds the output norm as $\|\rho v\|_{\infty} \leq T(\mathcal{C}) \cdot \|v\|_{\infty}$. When folding $K + k$ witnesses, the norm of the combined witness is bounded by $(K + k) \cdot T(\mathcal{C})$ times the maximum norm of the witnesses.

Proposition 2.16 (Expansion factors, [AL21]). *For any strong sampling set $\mathcal{C}$ over the cyclotomic ring $\mathbb{R}_{\mathbb{F}}$, the expansion factor $T(\mathcal{C})$ is strictly bounded by $2 \cdot \phi(\eta) \cdot \max_{\rho \in \mathcal{C}} \|\rho\|_{\infty}$.*

In practice, $\mathcal{C}$ is built using ring elements with coefficients drawn from a small interval $[-\beta, \beta]$. This gives expansion factor $T(\mathcal{C}) \approx \mathcal{O}(\phi(\eta) \cdot \beta)$, a modest constant, allowing the protocol to execute folding steps safely without norm blowup.

Remark 2.17 (The factor 2 is not needed for power-of-two cyclotomics). *Proposition 2.16 is stated for a general cyclotomic ring, and the factor 2 there accounts for the reduction modulo a general Φ . Under restriction (6) of §1.3 we work throughout with $\Phi(X) = X^d + 1$, d a power of two, and there the elementary bound is tight without it: multiplication in $\mathbb{R}_{\mathbb{F}}$ is a negacyclic convolution, so for any ρ, v*

$$\|\rho v\|_{\infty} \leq \|\rho\|_1 \|v\|_{\infty} \leq d\beta \|v\|_{\infty},$$

whence $T(\mathcal{C}) \leq \phi(\eta)\beta$.

Definition 2.18 (Special sets). Let $\mathcal{C}$ be a set and $\ell \in \mathbb{Z}_{\geq 1}$. For vectors $x, y \in \mathcal{C}^\ell$, define the relation $x \equiv_i y$ if and only if $x_i \neq y_i$ but $x_j = y_j$ for all $j \neq i$. A special set $\text{SS}(\mathcal{C}, \ell)$ is defined as:

$$\text{SS}(\mathcal{C}, \ell) = \{(\vec{c}, \vec{c}_1, \dots, \vec{c}_\ell) \in (\mathcal{C}^\ell)^{\ell+1} : \forall i \in [\ell], \vec{c} \equiv_i \vec{c}_i\}.$$

Lemma 2.19 (Coordinate-wise extraction, [FMN23, Lemma 7.1]). Let $\mathcal{C}$ be a finite set, $\ell \in \mathbb{N}$, and $\vec{\mathcal{C}} := \mathcal{C}^\ell$ be a challenge space. Let $\mathcal{A}: \vec{\mathcal{C}} \rightarrow \{0, 1\}^*$ be an arbitrary (probabilistic) EPT adversary, and $\mathcal{V}: \vec{\mathcal{C}} \times \{0, 1\}^* \rightarrow \{0, 1\}$ be an arbitrary PPT function (verification). Define the success probability of adversary $\mathcal{A}$ as $\epsilon^\mathcal{V}(\mathcal{A}) := \Pr [\mathcal{V}(\vec{c}, \mathcal{A}(\vec{c})) = 1]$. Then, there exists an EPT (oracle) extractor $\mathcal{E}_\mathcal{A}$ that makes at most $\ell + 1$ queries to $\vec{c} \xleftarrow{\$} \vec{\mathcal{C}}$ in expectation and with probability at least $\epsilon^\mathcal{V}(\mathcal{A}) - \frac{\ell}{|\vec{\mathcal{C}}|}$ outputs $\ell + 1$ pairs $(\vec{c}, w), (\vec{c}_1, w_1), \dots, (\vec{c}_\ell, w_\ell)$ such that

- $\mathcal{V}(\vec{c}, w) = 1$,
- for all $i \in [\ell]$, $\mathcal{V}(\vec{c}_i, w_i) = 1$, and
- $(\vec{c}, \vec{c}_1, \dots, \vec{c}_\ell) \in \text{SS}(\mathcal{C}, \ell)$.

In our protocols (mainly Protocol 7) the verifier samples coefficients from $\mathcal{C}$ rather than uniformly from $\mathbb{R}_\mathbb{F}$. This ensures that

- folded witnesses remain norm-bounded: $z_{\text{out}} = \sum_i \rho_i z_i$ satisfies $\|z_{\text{out}}\|_\infty < B = b^k$, and
- folded commitment salts remain binding-safe: $s_j = \sum_i \rho_i s_{i,j}$ satisfies $\|s_j\|_\infty < B_{\text{fold}}$.

Additionally, the relaxed-binding security of ABDLOP depends on adversarial differences being in $\mathcal{C} - \mathcal{C}$ (the subset of non-trivial differences in $\mathcal{C}$), which is exactly what arises from the verifier's sampling.

2.4. Recalls of SuperNeo. We recall here the SuperNeo [NS26] protocol. The protocol consists of a composition of reductions of knowledge $\Pi_{\text{SuperNeo}} := \Pi_{\text{DEC}} \circ \Pi_{\text{RLC}} \circ \Pi_{\text{CCS}}$, built on the global parameters (the fields, rings, and dimensions, the strong sampling set $\mathcal{C}$, the ring commitment scheme com , and the morphisms $\mathcal{L}, \mathcal{L}_{\text{in}}$) fixed once and for all in Definition 2.2 (§2.1).

The three reductions $\Pi_{\text{CCS}}, \Pi_{\text{RLC}}, \Pi_{\text{DEC}}$ are *not* our contribution, and none of them are modified in this subsection: we recall their relations and resulting properties in prose below, and refer the reader to [NS26, §§7.3–7.5] for the full protocol statements. Our own modifications to each of these three reductions, which constitute the technical core of the paper, are introduced later in §4; the reader already familiar with [NS26] can skip directly to §4 and use [NS26, §7] only as a lookup reference.

2.4.1. The embedding and the relations. As in [NS26, §7.1], we define a structure as a collection of elements

$$(2.1) \quad \mathbf{s} := \left\{ \{M_j \in \mathbb{F}^{m \times n_\mathbb{F}}\}_{j \in [t]}, f \in \mathbb{F}^{<u}[X_1, \dots, X_t] \right\},$$

which consists of matrices and a degree- u polynomial.

Let $M_j \in \mathbb{F}^{m \times n_\mathbb{F}}$ be a structural matrix. Following [NS26, §5], let us briefly recall the mechanism underlying the lift $M_j \mapsto \tilde{M}_j \in \mathbb{R}_{\mathbb{K}}^{m \times n_\mathbb{R}}$. Given a vector $v \in \mathbb{F}^d$, write $\mathbf{v} \in \mathbb{R}_\mathbb{F}$ for the ring element with coefficient vector $\text{cf}(\mathbf{v}) = v$; this *coefficient embedding* extends coordinate-wise to a field vector $z \in \mathbb{F}^{n_\mathbb{F}}$, partitioned into d -sized sub-vectors $z = [z_1, \dots, z_{n_\mathbb{R}}]$, by embedding each block, giving a ring vector $\mathbf{z} := (z_1, \dots, z_{n_\mathbb{R}}) \in \mathbb{R}_{\mathbb{F}}^{n_\mathbb{R}}$; and to a matrix $M \in \mathbb{F}^{m \times n_\mathbb{F}}$ with rows $M_1, \dots, M_m$ by embedding each row and stacking, giving $\mathbf{M} := \llbracket M_1, \dots, M_m \rrbracket \in \mathbb{R}_{\mathbb{F}}^{m \times n_\mathbb{R}}$.

Separately, SuperNeo also fixes a linear *inner product transform* $\bar{\cdot}: \mathbb{F}^d \rightarrow \mathbb{F}^d$ such that, for all $a, b \in \mathbb{F}^d$,

$$(2.2) \quad \text{ct}(\bar{a} \cdot b) = \langle a, b \rangle,$$

where ct denotes the constant coefficient (recall §2.1) and $\bar{a}$ denotes applying this transform to a before embedding it into $\mathbb{R}_\mathbb{F}$; that is, taking the constant term of the ring product of the *transformed-and-embedded* $\bar{a}$ against the plainly embedded b recovers the ordinary field inner product $\langle a, b \rangle$. Composing this transform with the coefficient embedding above, applied entrywise, is what turns a structural matrix $M \in \mathbb{F}^{m \times n_\mathbb{F}}$ into the object $\tilde{M}$ appearing in $\tilde{M}_j z(r)$. It is important to recall that, for $M \in \mathbb{F}^{m \times n_\mathbb{F}}$ and $z \in \mathbb{F}^{n_\mathbb{F}}$, the field matrix-vector product $Mz \in \mathbb{F}^m$ is recovered exactly as the vector of constant terms of $\tilde{M}\mathbf{z} \in \mathbb{R}_{\mathbb{F}}^m$, i.e. $Mz = \text{ct}(\tilde{M}\mathbf{z})$; all m row-wise inner products are thus computed *in parallel* by a single ring product rather than m separate field inner products. This is precisely what the evaluation claims $y_{i,j} = \tilde{M}_j z_i(r)$ of Definition 2.22 exploit: multiplying a ring element by an extension-field scalar in $\mathbb{K}$ scales each of its

d coefficients by that scalar, so the multilinear extension can be pushed inside coefficient-wise, and the constant term of the resulting evaluated ring vector, $\text{ct}(\widetilde{M_j \mathbf{z}}(r)) = \widetilde{M_j z}(r) \in \mathbb{K}$, is exactly the desired multilinear evaluation of the field vector $M_j z$ at the point r , batched over all m rows and computed via a single ring-valued Sum-Check rather than m separate field-valued ones. We also record, for later use, that the coefficient embedding is norm preserving, $\|\mathbf{v}\|_\infty = \|v\|_\infty$, so it does not by itself perturb the norm bounds tracked throughout §4.

Lemma 2.20 (Degree-0 ring elements multiply as scalars). *Let $\hat{a}, \hat{b} \in R_{\mathbb{K}}$ both be degree-0, i.e. $\text{cf}(\hat{a})_\ell = \text{cf}(\hat{b})_\ell = 0$ for all $\ell \geq 2$ (equivalently $\hat{a} = \text{ct}(\hat{a})$ and $\hat{b} = \text{ct}(\hat{b})$ as constant polynomials in $R_{\mathbb{K}}$). Then $\hat{a} \cdot \hat{b} \in R_{\mathbb{K}}$ is itself degree-0, and*

$$\text{ct}(\hat{a} \cdot \hat{b}) = \text{ct}(\hat{a}) \cdot \text{ct}(\hat{b}).$$

More generally, for any finite product $\hat{a}_1 \cdots \hat{a}_r$ of degree-0 elements, $\text{ct}(\hat{a}_1 \cdots \hat{a}_r) = \prod_{i=1}^r \text{ct}(\hat{a}_i)$.

Proof. Write $\hat{a} = \text{ct}(\hat{a}) \cdot 1$ and $\hat{b} = \text{ct}(\hat{b}) \cdot 1$ as elements of $R_{\mathbb{K}} = \mathbb{K}[X]/\Phi(X)$, i.e. constant polynomials. Multiplication in $R_{\mathbb{K}}$ restricted to the constant subring $\mathbb{K} \cdot 1 \subset R_{\mathbb{K}}$ agrees with multiplication in $\mathbb{K}$, since 1 is the multiplicative identity and $\Phi(X)$ has degree $d \geq 1$, so no reduction modulo $\Phi(X)$ is triggered by a product of two degree-0 polynomials. Hence $\hat{a} \cdot \hat{b} = (\text{ct}(\hat{a}) \cdot \text{ct}(\hat{b})) \cdot 1$, which is again degree-0 with constant term $\text{ct}(\hat{a}) \cdot \text{ct}(\hat{b})$. The general claim follows by induction on r . ■

The two main relations SuperNeo handles are the Norm-bounded CCS relation and the Norm-bounded CCS Evaluation relation. We note that [NS26] states these relations over a native relation field $\mathbb{L} = \mathbb{F}_{q^\tau}$ embedded, together with $\mathbb{K}$, into a common ambient field $\mathbb{A}$, with the witness packed into $R_{\mathbb{F}}^{n_{\mathbb{R}}}$ via $\text{Flat}_{\mathbb{L}}$, $\text{Pack}_{\mathbb{L}}$ and $\text{Trans}_{\mathbb{L}}$ ([NS26, §5]). We work throughout at $\tau = 1$, i.e. $\mathbb{L} = \mathbb{F}$ and $\mathbb{A} = \mathbb{K}$, as in all three parameterizations of [NS26, §8], and state the relations over $\mathbb{F}$ accordingly; see [NS26, Definitions 19–20] for the general form.

Definition 2.21 (Norm-bounded CCS). *Let $\mathcal{L} : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow \mathbb{C}$ be an arbitrary $R_{\mathbb{F}}$ -module homomorphism. Let $\mathbf{s}$ be a structure as defined above. We define the norm-bounded CCS relation, $\text{CCS}(b, \mathcal{L})$, as follows:*

$$\left\{ \begin{array}{l} \text{For } \mathbf{z} := [x, w], \\ (\mathbf{s}; (c \in \mathbb{C}, x \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}); w \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}, \text{in}}}) : c = \mathcal{L}(\mathbf{z}) \wedge \|\mathbf{z}\|_\infty < b \wedge \\ f(\widetilde{M_1 \mathbf{z}}, \dots, \widetilde{M_t \mathbf{z}}) \in \text{ZS}_{\log_2(m)} \end{array} \right\}$$

Definition 2.22 (Norm-bounded CCS Evaluation Relation). *Let $\mathbf{s}$ be a structure as defined above. Let $\mathcal{L} : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow \mathbb{C}$ be an arbitrary $R_{\mathbb{F}}$ -module homomorphism. Define $\mathcal{L}_{\text{in}} : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow R_{\mathbb{F}}^{n_{\mathbb{R}, \text{in}}}$ to be the trivial $R_{\mathbb{F}}$ -module homomorphism that projects the first $n_{\mathbb{R}, \text{in}}$ indices. We define the **norm-bounded CCS evaluation relation**, $\text{CE}(b, \mathcal{L})$, as follows:*

$$\left\{ \left(\mathbf{s}; \left(\begin{array}{l} c \in \mathbb{C}, \\ x \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}, \\ r \in \mathbb{K}^{\log_2(m)}, \\ \{y_j \in R_{\mathbb{K}}\}_{j \in [t]} \end{array} \right); z \in \mathbb{F}^{n_{\mathbb{F}}} \right) : \begin{array}{l} c = \mathcal{L}(\mathbf{z}) \wedge \mathbf{x} = \mathcal{L}_{\text{in}}(\mathbf{z}) \\ \wedge \|\mathbf{z}\|_\infty < b \wedge \\ \forall j \in [t], y_j = \widetilde{M_j \mathbf{z}}(r) \end{array} \right\}$$

Remark 2.23. *The morphism $\mathcal{L}$ is taken to be a ring commitment scheme (see §2.5) $\text{Commit}(\text{pi}, \bullet)$; recall Definition 2.2 and cf. [NS26, Definition 21].*

2.4.2. The interactive reductions. The full protocol statements may be found in [NS26, §§7.3–7.5]; we record here only a brief account of them and the properties we rely on.

2.4.2.1. CCS Reduction – Π_{CCS} . The main interactive reduction we need to address in order to define a suitable LatticeBlindFold protocol is the RoK Π_{CCS} . Given K fresh CCS instances and k prior evaluation claims, it checks (simultaneously) that the K CCS instances satisfy their constraints, that the k evaluation claims hold at the carried-over point r , and that the norms of all $K + k$ witness vectors involved are bounded by b ; the output is a fresh batch of $K + k$ evaluation claims at a new point r' .

It achieves this via a single invocation of the Sum-Check protocol on a composite polynomial

$$(2.3) \quad Q(\vec{X}) := \text{eq}(\vec{X}, \alpha) \cdot (F(\vec{X}) + \gamma^K \cdot \text{NC}(\vec{X})) + \gamma^{2K+k} \cdot \text{Eval}(\vec{X}) \in \mathbb{K}[\vec{X}],$$

where $F(\vec{X})$ encodes the K CCS constraints, $NC(\vec{X})$ encodes the $K + k$ norm constraints, $Eval(\vec{X})$ encodes the k carried-over evaluation claims, and α, γ are verifier challenges batching these three families of checks into one sum. The prover sends fresh evaluation hints $y'_{i,j} := \widetilde{M_j z_i}(r')$ at the end of the Sum-Check, and the verifier checks a single evaluation identity combining them.

Lemma 2.24 (Π_{CCS} is strong, [NS26, Lemma 7, Appendix B.2]). *The interactive reduction $\Pi_{\text{CCS}} : \text{CCS}(b, \mathcal{L})^K \times \text{CE}(b, \mathcal{L})^k \rightarrow \text{CE}(b, \mathcal{L})^{K+k}$ is strong for the function ϕ , which projects commitments $(c_i)_{i \in [K+k]}$ from the instance.*

2.4.2.2. Random linear combination reduction – Π_{RLC} . The interactive reduction Π_{RLC} performs the explicit folding step: given $K + k$ input CCS evaluation claims of norm b , the verifier samples challenges $\rho_1, \dots, \rho_{K+k} \leftarrow \mathcal{C}$ from the strong sampling set (Definition 2.13), and the prover and verifier jointly compute the corresponding random linear combination of commitments, public inputs, and evaluation hints, outputting a single claim of larger norm $B = b^k$.

Lemma 2.25 (Π_{RLC} is weak, [NS26, Lemma 8, Lemma 11, Appendix B.3]). *The interactive reduction $\Pi_{\text{RLC}} : \text{CE}(b, \mathcal{L})^{K+k} \rightarrow \text{CE}(B, \mathcal{L})$ is **weak** for the function ϕ , which projects commitments $(c_i)_{i \in [K+k]}$ from the instance.*

2.4.2.3. Decomposition reduction – Π_{DEC} . The final reduction reduces the norm of claims from $B = b^k$ back down to b , so that CCS claims can be folded repeatedly without the norm of the witness openings growing without bound. The prover computes a b -ary decomposition $\text{split}_b(\mathbf{z}) = (\mathbf{z}_1, \dots, \mathbf{z}_k)$ of the witness (Definition 2.2), together with matching decompositions of the commitment and evaluation hints, and the verifier checks that these recombine correctly.

Proposition 2.26 (Π_{DEC} is a RoK, [NS26, Theorem 13]). *$\Pi_{\text{DEC}} : \text{CE}(B, \mathcal{L}) \rightarrow \text{CE}(b, \mathcal{L})^k$ is a reduction of knowledge.*

2.5. Ring commitments. We consider several commitment schemes, all over rings.

Definition 2.27 (Ring Commitment Scheme, [NS26, Definition 7]). *A ring commitment scheme $\text{com} := (\text{Setup}, \text{Commit})$ consists of two PPT algorithms:*

- **Setup** $(1^\lambda, m) \rightarrow \text{pp}$: Takes as input a security parameter 1^λ and length $m \in \mathbb{Z}_{\geq 1}$, outputs public parameters pp .
- **Commit** $(\text{pp}, z) \rightarrow c$: Takes as input public parameters pp and a vector $z \in R_{\mathbb{F}}^m$, outputs a commitment $c \in \mathcal{C}$.

A ring commitment scheme can satisfy the following properties:

- **B -binding**: For every length $m = \text{poly}(\lambda)$ and every EPT adversary $\mathcal{A}$, a ring commitment scheme is B -binding (for $B \in \mathbb{Z}_{\geq 1}$) if the following probability holds:

$$\Pr \left[\begin{array}{l} \text{Commit}(\text{pp}, z_1) = \text{Commit}(\text{pp}, z_2) \\ \wedge \|z_1\|_\infty, \|z_2\|_\infty < B, \\ \wedge z_1 \neq z_2 \end{array} \middle| \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda, n_{\mathbb{R}}) \\ z_1, z_2 \in R_{\mathbb{F}}^m \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \leq \epsilon_{\text{bind}}(B)$$

for $\epsilon_{\text{bind}}(B) \leq \text{negl}(\lambda)$. We refer to a pair of vectors (z_1, z_2) which satisfies the conditions in the probability as a B -binding collision.

- **$(B, \mathcal{C})$ -relaxed binding**: For every length $m = \text{poly}(\lambda)$ and every EPT adversary $\mathcal{A}$, a ring commitment scheme is $(B, \mathcal{C})$ -relaxed binding (for $B \in \mathbb{Z}_{\geq 1}$ and set $\mathcal{C} \subseteq R_{\mathbb{F}}$) if the following probability holds:

$$\Pr \left[\begin{array}{l} \Delta_1 \cdot c = \text{Commit}(\text{pp}, z_1) \\ \wedge \Delta_2 \cdot c = \text{Commit}(\text{pp}, z_2) \\ \wedge \|z_1\|_\infty, \|z_2\|_\infty < B, \\ \wedge \Delta_1 z_2 \neq \Delta_2 z_1 \end{array} \middle| \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda, m) \\ c \in \mathcal{C}, \\ \left(\begin{array}{l} \Delta_1, \Delta_2 \in (\mathcal{C} - \mathcal{C}), \\ z_1, z_2 \in R_{\mathbb{F}}^m \end{array} \right) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \leq \epsilon_{\text{rlx}}(B, \mathcal{C})$$

for $\epsilon_{\text{rlx}}(B, \mathcal{C}) \leq \text{negl}(\lambda)$. We refer to a tuple of elements $(c, \Delta_1, \Delta_2, z_1, z_2)$ which satisfies the conditions in the probability as a $(B, \mathcal{C})$ -relaxed binding collision.

- **Homomorphic**: For every $m \in \mathbb{Z}_{\geq 1}$ and $\text{pp} \in \text{Setup}(1^\lambda, m)$, the commitment algorithm, $\text{Commit}(\text{pp}, \cdot) : R_{\mathbb{F}}^m \rightarrow \mathcal{C}$, is an $R_{\mathbb{F}}$ -module homomorphism.

2.5.1. Ajtai Commitment Schemes. We begin by introducing the Ajtai commitment scheme, which serves as the foundational, succinct, lattice-based commitment protocol used throughout our folding constructions. Relying on the Module Short Integer Solution (MSIS) assumption, the Ajtai scheme provides binding, homomorphic commitments for norm-bounded ring elements.

Definition 2.28 (Compact Ajtai commitment scheme). *Let the message length be $m \in \mathbb{Z}_{\geq 1}$ and let $\kappa \in \mathbb{Z}_{\geq 1}$ denote the commitment rank. The Ajtai commitment scheme $\text{com}_{\text{Ajtai}} := (\text{Setup}, \text{Commit})$ over a ring $\mathbb{R}_{\mathbb{F}}$ consists of two PPT algorithms:*

- $\text{Setup}(1^\lambda, m, \kappa) \rightarrow \text{pp}$: Samples a uniformly random matrix $A \xleftarrow{\$} \mathbb{R}_{\mathbb{F}}^{\kappa \times m}$. Outputs public parameters $\text{pp} := A$.
- $\text{Commit}(\text{pp}, z) \rightarrow c$: Given pp and a vector $z \in \mathbb{R}_{\mathbb{F}}^m$, outputs the commitment $c := Az \in \mathbb{R}_{\mathbb{F}}^\kappa$.

Remark 2.29. A critical property of $\text{com}_{\text{Ajtai}}$ is that it is perfectly homomorphic over $\mathbb{R}_{\mathbb{F}}$: $\text{Commit}(\text{pp}, z_1 + \alpha z_2) = \text{Commit}(\text{pp}, z_1) + \alpha \text{Commit}(\text{pp}, z_2)$. Note, however, that standard Ajtai commitments (as well as the ones considered in [NS26]) are not hiding. For components of our protocol requiring zero-knowledge (e.g., hiding witnesses or evaluation hints), we will later introduce the computationally hiding ABDLOP commitment schemes (see §2.5.2).

Proposition 2.30 ([NS26, Theorem 6]). *The Ajtai commitment scheme is a ring commitment scheme that is homomorphic B -binding (assuming the $\text{MSIS}_{m,2B}^{\infty,\kappa,q}$ problem is hard), and $(B, \mathcal{C})$ -relaxed binding (assuming the $\text{MSIS}_{m,4TB}^{\infty,\kappa,q}$ problem is hard) and $\mathcal{C}$ is a strong sampling set with expansion factor T .*

2.5.2. The ABDLOP commitment scheme. We need to consider a lattice-based homomorphic commitment so that we can commit to generic ring vectors without norm constraints. Let Commit be such commitment scheme. More precisely, we consider the ABDLOP [LNP22] commitment scheme (an improvement on BDLOP [BDL⁺16]); we briefly recall here the form of such commitments.

Let $A_1 \in \mathbb{R}_{\mathbb{F}}^{\kappa \times m_1}$ and $A_2 \in \mathbb{R}_{\mathbb{F}}^{\kappa \times m_2}$ be two matrices, uniformly random, where κ plays the same role as the number of rows of the Ajtai commitment matrix recalled above; in the ABDLOP [LNP22] we sample also a third public matrix $B \in \mathbb{R}_{\mathbb{F}}^{\ell \times m_2}$, where $\ell = \kappa'$ is the number of (unrestricted) ring elements we want to commit to. Let $y \in \mathbb{R}_{\mathbb{F}}^\ell$ be the message we want to commit to. Given random vectors of small norm (in [LNP22] these are sampled with bounded ℓ_2 -norm, but, to stay consistent with the ℓ_∞ -norm convention fixed earlier in this section, we sample them with bounded ℓ_∞ -norm instead, which loosens the resulting bounds by a factor of at most $\sqrt{m_1 d}$ for a vector $s_1 \in \mathbb{R}_{\mathbb{F}}^{m_1}$, since $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{|x|} \|x\|_\infty$ with $|x| = m_1 d$ the length of x counted in field elements; cf. §3.3.4, and note that the factor is $\sqrt{d}$ only for a *single* ring element) say $s_1 \in \mathbb{R}_{\mathbb{F}}^{m_1}$ with $\|s_1\|_\infty < \alpha$ and $s_2 \in \mathbb{R}_{\mathbb{F}}^{m_2}$ with $\|s_2\|_\infty < \nu$, we define the two commitments as

$$(2.4) \quad \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \pmod{q} = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \cdot s_1 + \begin{pmatrix} 0^n \\ y \end{pmatrix} \quad =: \text{Commit}_{\text{BDLOP}}(m),$$

$$(2.5) \quad \begin{pmatrix} t_A \\ t_B \end{pmatrix} \pmod{q} = \begin{pmatrix} A_1 \\ 0 \end{pmatrix} \cdot s_1 + \begin{pmatrix} A_2 \\ B \end{pmatrix} \cdot s_2 + \begin{pmatrix} 0^n \\ y \end{pmatrix} \quad =: \text{Commit}_{\text{ABDLOP}}(m).$$

ABDLOP [LNP22] is natively developed over primes $q \equiv 5 \pmod{8}$, similarly to [NS24].

Now, considering any ring element as a field vector via the identification provided by cf , we can apply these commitment schemes to immediately deduce the following lemma.

Lemma 2.31. *Recall the notation ct and cf introduced in §2.4. Let $y \in \mathbb{R}_{\mathbb{F}}^\ell$ be a committed message. Fixed the randomness (s_1, s_2) and the public matrices (A_1, A_2, B) we have that:*

$$(2.6) \quad \text{cf}(\text{Commit}_{\text{ABDLOP}}(y)) := \begin{pmatrix} \text{cf}(t_A) \\ \text{cf}(t_B) \end{pmatrix} \pmod{q} = \begin{pmatrix} \text{cf}(A_1 \cdot s_1) + \text{cf}(A_2 \cdot s_2) \\ \text{cf}(B \cdot s_2) + \text{cf}(y) \end{pmatrix}.$$

Therefore,

$$(2.7) \quad (\text{cf}(\text{Commit}_{\text{ABDLOP}}(y)))_\ell := \begin{pmatrix} (\text{cf}(t_A))_\ell \\ (\text{cf}(t_B))_\ell \end{pmatrix} \pmod{q} = \begin{pmatrix} (\text{cf}(A_1 \cdot s_1))_\ell + (\text{cf}(A_2 \cdot s_2))_\ell \\ (\text{cf}(B \cdot s_2))_\ell + (\text{cf}(y))_\ell \end{pmatrix},$$

for each ℓ . Recall that for $\ell = 1$, $\text{cf}(\bullet)_1 = \text{ct}$.

The same is true for $\text{Commit}_{\text{BDLOP}}$.

Together with the property of Lemma 2.31, a key motivation for the adoption of the ABDLOP commitment scheme is its zero-knowledge proof of knowledge (ZKPoK): the commitment scheme is endowed with an argument of knowledge for verifying that the prover knows (unique) salts (s_1, s_2) , together with a simulator which proves the argument is zero-knowledge.

2.6. Security assumptions. The underlying security of our folding procedure fundamentally relies on well-established lattice hardness assumptions defined over cyclotomic rings. Let $\mathcal{R}$ denote a cyclotomic ring, and let $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ for a chosen modulus q . The central assumption required for knowledge soundness in this context is the Module Short Integer Solution (MSIS) problem [LS15, BDL⁺16, LNP22].

Definition 2.32 (Module-SIS (MSIS _{n,m,q,B})). *Given a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$, the MSIS _{n,m,q,B} problem asks to find a non-zero vector $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \pmod{q}$ and $\|\mathbf{z}\| \leq B$.*

Unless subscripted, $\|\cdot\|$ denotes the ℓ_∞ -norm fixed earlier in this section, extended coefficient-wise to vectors and matrices of ring elements.

As observed in [NS26] and [BC24], strictly bounding the norm B across sequential operations is critical. If the norms grow too rapidly during folding, the scheme becomes vulnerable to trivial attacks or requires impractically large parameters to preserve the MSIS assumption.

Definition 2.33 (Module-LWE, decisional (MLWE _{n,m,q,χ})). *Let χ be a distribution over $\mathcal{R}$ (concretely, throughout this paper, the uniform distribution over $\{c \in \mathcal{R} : \|c\|_\infty \leq \nu\}$ for a bound ν). Given a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$ and a secret $\mathbf{s} \leftarrow \chi^m$, the MLWE _{n,m,q,χ} problem asks to distinguish $(\mathbf{A}, \mathbf{A}\mathbf{s} \bmod q)$ from $(\mathbf{A}, \mathbf{u})$ for $\mathbf{u} \leftarrow \mathcal{R}_q^n$ uniform, with non-negligible advantage.*

Definition 2.34 (Extended MLWE _{$m',k',\ell',\chi,\xi_1,\xi_2$} [LNS20, Definition 3.3]). *Let us denote by $\langle -, - \rangle$ the usual Euclidean inner product over vectors with coefficients in the ring $\mathcal{R}$. The Extended Module-LWE problem with parameters $m', \ell' > 0$ and error distributions χ, ξ_1 and ξ_2 over $\mathcal{R}$ and $\mathcal{R}^{k'}$ respectively, asks the adversary $\mathcal{A}$ to distinguish between the following two cases:*

- (1) $\left(\mathbf{B}, \mathbf{B}\vec{\mathbf{r}}, \mathbf{c}_1, \dots, \mathbf{c}_{k'}, \vec{\mathbf{z}}, \text{sign} \left(\left\langle \vec{\mathbf{z}}, \begin{pmatrix} \mathbf{c}_1 \vec{\mathbf{r}} \\ \vdots \\ \mathbf{c}_{k'} \vec{\mathbf{r}} \end{pmatrix} \right\rangle \right) \right)$ for $\mathbf{B} \leftarrow \mathcal{R}_q^{m' \times (m' + \ell')}$, a secret vector $\vec{\mathbf{r}} \leftarrow \chi^{m' + \ell'}$ and $(\vec{\mathbf{z}}, \mathbf{c}_1, \dots, \mathbf{c}_{k'}) \leftarrow \xi_1^{k'(m' + \ell')} \times \xi_2$
- (2) $\left(\mathbf{B}, \vec{\mathbf{u}}, \mathbf{c}_1, \dots, \mathbf{c}_{k'}, \vec{\mathbf{z}}, \text{sign} \left(\left\langle \vec{\mathbf{z}}, \begin{pmatrix} \mathbf{c}_1 \vec{\mathbf{r}} \\ \vdots \\ \mathbf{c}_{k'} \vec{\mathbf{r}} \end{pmatrix} \right\rangle \right) \right)$ for $\mathbf{B} \leftarrow \mathcal{R}_q^{m' \times (m' + \ell')}$, $\vec{\mathbf{u}} \leftarrow \mathcal{R}_q^{m'}$, a secret vector $\vec{\mathbf{r}} \leftarrow \chi^{m' + \ell'}$ and $(\vec{\mathbf{z}}, \mathbf{c}_1, \dots, \mathbf{c}_{k'}) \leftarrow \xi_1^{k'(m' + \ell')} \times \xi_2$,

where $\text{sign}(a) = 1$ if $a \geq 0$ and 0 otherwise. Then, $\mathcal{A}$ is said to have advantage ϵ in solving Extended MLWE _{$m',k',\ell',\chi,\xi_1,\xi_2$} if

$$\left| \Pr \left[b = 1 \mid \mathbf{B} \leftarrow \mathcal{R}_q^{m' \times (m' + \ell')}; \vec{\mathbf{r}} \leftarrow \chi^{m' + \ell'}; \vec{\mathbf{z}} \leftarrow \xi_1^{k'(m' + \ell')}; \vec{\mathbf{c}} \leftarrow \xi_2; b \leftarrow \mathcal{A}(\mathbf{B}, \mathbf{B}\vec{\mathbf{r}}, \vec{\mathbf{z}}, \vec{\mathbf{c}}, s) \right] \right. \\ \left. - \Pr \left[b = 1 \mid \mathbf{B} \leftarrow \mathcal{R}_q^{m' \times (m' + \ell')}; \vec{\mathbf{u}} \leftarrow \mathcal{R}_q^{m'}; \vec{\mathbf{r}} \leftarrow \chi^{m' + \ell'}; \vec{\mathbf{z}} \leftarrow \xi_1^{k'(m' + \ell')}; \vec{\mathbf{c}} \leftarrow \xi_2; b \leftarrow \mathcal{A}(\mathbf{B}, \vec{\mathbf{u}}, \vec{\mathbf{z}}, \vec{\mathbf{c}}, s) \right] \right| \geq \epsilon.$$

where

$$s = \text{sign} \left(\left\langle \vec{\mathbf{z}}, \begin{pmatrix} \mathbf{c}_1 \vec{\mathbf{r}} \\ \vdots \\ \mathbf{c}_{k'} \vec{\mathbf{r}} \end{pmatrix} \right\rangle \right) \quad \text{and } \vec{\mathbf{c}} = (\mathbf{c}_1, \dots, \mathbf{c}_{k'}).$$

The hiding of $\mathcal{L}$ under the blinded layout (Lemma 3.3) is, in its default regime, unconditional ([BDL⁺16, Proof of Lemma 4] uses only the Leftover Hash Lemma). Its computational fallback rests on DKS[∞] (Definition 2.35, [BDL⁺16, Definition 2]). Relaxed binding of $\mathcal{L}$ remains MSIS in both regimes; cf. Remark 3.4.

Definition 2.35 (Decisional Knapsack, DKS _{n,k,β} [∞], [BDL⁺16]). *For $k > n$ and $\beta \geq 1$ write $S_\beta := \{c \in \mathcal{R} : \|c\|_\infty < \beta\}$. The DKS _{n,k,β} [∞] problem asks to distinguish*

$$(\mathbf{A}', [\mathbf{I}_n \mid \mathbf{A}']\mathbf{y}), \quad \mathbf{A}' \leftarrow \mathcal{R}_q^{n \times (k-n)}, \mathbf{y} \leftarrow S_\beta^k,$$

from $(\mathbf{A}', \mathbf{u})$ with $\mathbf{u} \leftarrow \mathcal{R}_q^n$ uniform.

Remark 2.36 (Security of the constructions recalled above). *Every hardness assumption used in this paper reduces to MSIS (Definition 2.32) or MLWE (Definitions 2.33–2.34) over $\mathbb{R}_{\mathbb{F}}$ ($= \mathcal{R}$), both believed to resist quantum adversaries: under MSIS the Ajtai and $\text{Commit}_{\text{BDLOP}}$ / $\text{Commit}_{\text{ABDLOP}}$ schemes of §2.5 are (relaxed-)binding, and under MLWE the latter two are additionally hiding, with honest-verifier zero-knowledge proofs of knowledge.*

In the end, as in [LNP22, LNS20], we assume that Extended-MLWE is almost as hard as plain MLWE. Similarly, we do assume that the hardness of $\text{DKS}_{n,k,\beta}^{\infty}$ is the same of MLWE at secret dimension $k - n$ with n samples.

3. THE ZK-FYING COMPONENTS

Having recalled in §2 the interactive-reduction framework, the notion of a blinding folding reduction, and the commitment and hardness assumptions we rely on, we now turn to the technical core of the paper, that is, making SuperNeo’s Sum-Check reduction zero-knowledge, one obstruction at a time.

3.1. Hiding for Ajtai commitments. The one place where an Ajtai image must nevertheless be hiding is the input instance $c_1 = \mathcal{L}(\mathbf{z}_1)$, which no ABDLOP layer can wrap without destroying succinctness of the folded instance. We handle it by randomizing the *preimage* instead via Lemma 3.3. We make use of Definition 3.1.

Definition 3.1 (Blinded R1CS layout). *Fix $n_{\mathbb{F},\text{bl}}$ as in §2.1 and let $\mathcal{I}_{\text{bl}} := \{n_{\mathbb{F}} - n_{\mathbb{F},\text{bl}} + 1, \dots, n_{\mathbb{F}}\}$. We say that a structure $\mathbf{s} = (M_1, \dots, M_t, f)$ and a witness follow the blinded layout if*

- (1) $\mathbf{z} = [x, w]$ exactly as in Definition 2.21, with the last $n_{\mathbb{F},\text{bl}}$ coordinates of w , indexed by $\mathcal{I}_{\text{bl}}$, designated as the blinding block e and sampled uniformly from $\{v : \|v\|_{\infty} < b\}$ by the honest prover.
- (2) some fixed index $\iota_1 \leq n_{\mathbb{F},\text{in}}$ carries the constant-one wire, $z_{\iota_1} = 1$;
- (3) for every $i \in \mathcal{I}_{\text{bl}}$, $(M_1)_{i,\bullet} = \delta_i$, $(M_2)_{i,\bullet} = \delta_i$ and $(M_3)_{i,\bullet} = \delta_{\iota_1}$, where δ_j is the j -th standard basis row;
- (4) for every $i \notin \mathcal{I}_{\text{bl}}$ and every $j \in [t]$, $(M_j)_{i,\mathcal{I}_{\text{bl}}} = 0$: the circuit rows have no support on the blinding columns.

Clause (4) holds automatically when the layout is reached by appending fresh variables and fresh rows to an existing structure, which is how it arises in practice. We state it because the truncation of Lemma 3.29 (see below) depends on it, and clauses (1)–(3) alone do not imply it. Note that it constrains the constraint matrices only: the commitment $\mathcal{L}$ does have support on $\mathcal{I}_{\text{bl}}$, and must, since $A_{\text{bl}}e$ is what Lemma 3.3 relies on.

Remark 3.2. *Under Definition 3.1, row $i \in \mathcal{I}_{\text{bl}}$ of the R1CS predicate $f(Y_1, Y_2, Y_3) = Y_2 Y_3 - Y_1$ reads $z_i \cdot 1 - z_i = 0$, which holds for every value of z_i . The block is therefore unconstrained by $\mathbf{s}$, and the language decided by $\text{CCS}(b, \mathcal{L})$ is unchanged. A circuit is satisfiable for public input x under the blinded layout iff it is under the original one.*

Now, it follows that we can still assume that $M_1 = \mathbf{I}_m, \text{NC}(\vec{X}) = \sum_i \gamma_i^{(1)} \prod_j (\tilde{z}_i - j)$ ranges over all m coordinates, so $\|e\|_{\infty} < b$ is certified by the existing norm check with no new machinery, and $\mathcal{L}$ is the same matrix, still an $\mathbb{R}_{\mathbb{F}}$ -module homomorphism, and e travels through Π'_{RLC} ’s rejection sampling and Π'_{DEC} ’s split_b as ordinary witness coordinates.

Lemma 3.3 ($\mathcal{L}$ is hiding under the blinded layout). *Write $\mathcal{L}(\mathbf{z}) = A_{\text{circ}} \mathbf{z}_{\text{circ}} + A_{\text{bl}} \mathbf{e}$ for the column splitting induced by $\mathcal{I}_{\text{bl}}$, and let χ_b be uniform on $\{c \in \mathbb{R}_{\mathbb{F}} : \|c\|_{\infty} \leq b - 1\}$. Then for every $\mathbf{z}_{\text{circ}}$, the distribution of $\mathcal{L}([\mathbf{z}_{\text{circ}}, \mathbf{e}])$ over $\mathbf{e} \leftarrow \chi_b^{n_{\text{R},\text{bl}}}$ is within $\varepsilon_{\mathcal{L}\text{-hide}}$ of uniform on $\mathbb{C}$, in either regime:*

- (1) (Statistical.) *If the cyclotomic polynomial $X^d + 1$ splits into l irreducible factors modulo q , if $q \equiv 2l + 1 \pmod{4l}$, and if*

$$q^{\kappa/n_{\text{R},\text{bl}}} \cdot 2^{256/(n_{\text{R},\text{bl}} d)} \leq 2(b - 1) < \mathbf{b}_{\text{inv}},$$

then $\varepsilon_{\mathcal{L}\text{-hide}} \leq 2^{-128}$, statistically and against an unbounded distinguisher. Here $\kappa = \kappa_{\text{Ajtai-Compact}}$.

- (2) (Computational.) *Otherwise, if the decisional knapsack problem $\text{DKS}_{\kappa, \kappa + n_{\text{R},\text{bl}}, b}^{\infty}$ is hard, the bound holds computationally with $\varepsilon_{\mathcal{L}\text{-hide}}$ the corresponding advantage.*

Proof. $A_{\text{circ}} \mathbf{z}_{\text{circ}}$ is a fixed shift and the uniform distribution on $\mathbb{C}$ is invariant under it, so it suffices that $A_{\text{bl}} \mathbf{e}$ be uniform, resp. pseudorandom; A_{bl} is a column block of the uniform Ajtai matrix, hence uniform. Regime (1) is [BDL⁺16, Lemma 4] applied to A_{bl} , with their ring degree N read as our d , their number of splitting factors d as our l , and $\beta = b - 1$. The upper bound is exactly the invertibility threshold $\mathbf{b}_{\text{inv}}$ of Proposition 2.14. Regime (2) is [BDL⁺16, Definition 2], after replacing A_{bl} by its Hermite normal form: fixing the last κ columns and left-multiplying by their inverse is a bijection of $\mathcal{R}_q^\kappa$ and so preserves any advantage, and those columns are invertible except with probability at most $l/(q^{d/l} - 1)$. In that form $[\mathbf{I}_\kappa \mid A'] \mathbf{y} = A' \mathbf{y}_1 + \mathbf{y}_2$ is normal-form MLWE with secret dimension $n_{\text{R,bl}}$ and κ samples, as hard as the uniform-secret variant by [ACPS09] and [LS15]. ■

We stress that the layout by itself is not enough. If the security experiment is read over the already-padded relation, the adversary supplies the blinding block along with the rest of the witness and may set $\mathbf{e} := 0$; the segment is then as deterministic as everything else and Lemma 3.3 has no hypothesis to work with. What the layout buys is the *possibility* of relocating the entropy; what makes it available is reading the experiment over the unpadded relation of Definition 3.28, where $\mathcal{A}$ supplies only $(x_1, \mathbf{w}_{\text{circ}})$ and $\mathbf{e}$ is drawn afterwards by the prover as part of the embedding (3.19). It is in that reading, and only in it, that the lemma is invoked below; cf. Remark 3.27.

Remark 3.4 (Binding is untouched). *Statistical hiding on the blinding block neither weakens nor interacts with the (B,C)-relaxed binding of Definition 2.27. Unfolding that definition, a relaxed collision $(c, \Delta_1, \Delta_2, z_1, z_2)$ gives $\mathcal{L}(\Delta_2 z_1 - \Delta_1 z_2) = \Delta_2 \Delta_1 c - \Delta_1 \Delta_2 c = 0$ with $\Delta_2 z_1 - \Delta_1 z_2 \neq 0$ of norm below $4TB$, i.e. an MSIS solution. Only the message length changes, $\text{MSIS}_{n_{\text{R}}, 4TB}^{\infty, \kappa, q} \rightarrow \text{MSIS}_{n_{\text{R}} + n_{\text{R,bl}}, 4TB}^{\infty, \kappa, q}$. Note that even though the size is different, the bound $B = b^k$ is unchanged, since $\|\mathbf{e}\|_\infty < b$ and the block folds as an ordinary witness segment. What the block does foreclose is a statistically binding instantiation of $\mathcal{L}$, in the sense available to [BDL⁺16] for their own scheme. However, nothing here or in [NS26] needs it ($\mathcal{L}$ compresses $\mathcal{R}_q^{n_{\text{R}}}$ onto $\mathcal{R}_q^\kappa$) and the binding was already only computationally binding.*

3.2. Companion procedures from the ABDLOP proofs of knowledge. One main feature of the ABDLOP commitment scheme that we make use of is that it is endowed with zero-knowledge proof of knowledge of randomness and messages as well as of their linear and quadratic relations. We do not report them in full and simply recall the main results we need (together with a small modification). We refer the reader to the algorithms in [LNP22, Fig. 4, Fig. 5] for the linear relations, and to [LNP22, §4, Fig. 6] for the quadratic-relation proof system. As in loc. cit. we fix $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d + 1)$ to be the cyclotomic ring modulo q , and let $S_{\beta_{\text{ch}}} = \{x \in \mathcal{R}_q : \|x\|_\infty \leq \beta_{\text{ch}}\}$ be the set of polynomials with coefficients bounded by β_{ch} , which acts as a selection set for the strong sampling set used in these proofs.

Assumption 3.5. *For the sake of simplicity, we do not track their completeness errors, as they are intertwined with their rejection parameter; cf. [LNP22, Lemma 2.14, Theorem 4.2, Theorem 4.5, Figure 12]. Instead, we simply assume that they are negligible in the security parameter λ_{ABDLOP} , and fix them to be $2^{-\lambda_{\text{ABDLOP}}} \leq 2^{-\lambda}$.*

Lemma 3.6. *Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ [LNP22, Fig. 4] is a proof of knowledge of $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \bar{\mathcal{C}}$ satisfying*

- (1) $\mathbf{A}_1 \bar{\mathbf{s}}_1 + \mathbf{A}_2 \bar{\mathbf{s}}_2 = \mathbf{t}_A$,
- (2) $\|\bar{\mathbf{s}}_i \bar{c}\| \leq 2\mathbf{s}_i \sqrt{2m_i d}$ for $i = 1, 2$, and
- (3) $\mathbf{R}_1 \bar{\mathbf{s}}_1 + \mathbf{R}_m(\mathbf{t}_B - \mathbf{B} \bar{\mathbf{s}}_2) = \mathbf{u}$.

Under the assumption that $\text{MSIS}_{n, m_1 + m_2, B}$ is hard for $B = 8\eta \sqrt{(\mathbf{s}_1 \sqrt{2m_1 d})^2 + (\mathbf{s}_2 \sqrt{2m_2 d})^2}$, then

- *the pair $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2)$ is unique, and*
- *for any two valid transcripts $(\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2)$ and $(\mathbf{w}, \mathbf{v}, c', \mathbf{z}'_1, \mathbf{z}'_2)$, it holds that $\mathbf{z}_i - c\bar{\mathbf{s}}_i = \mathbf{z}'_i - c'\bar{\mathbf{s}}_i$.*

Moreover, the protocol admits a simulator $\mathcal{S}_{\text{ABDLOP}}$ such that, without access to private information $\mathbf{s}_1, \mathbf{s}_2$, and $\mathbf{m}$, outputs a simulation of a commitment $(\mathbf{t}_A, \mathbf{t}_B)$ along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has non-negligible (depending on ε) advantage in distinguishing the Extended-MLWE $_{n+\ell, 1, m_2 - n - \ell, \chi, \mathcal{C}, \mathbf{s}_2}$.

Proof. The first three statements are [LNP22, Lemma 3.1]. Regarding the simulator $\mathcal{S}_{\text{ABDLOP}}$, we report here the explicit hybrid construction contained in the proof of [LNP22, Theorem 4.2]; we refer the reader to loc. cit. for all the details. Here we only sketch the construction as we will make use of these simulators in our proofs. Following [LNP22, Theorem 4.2], we build $\mathcal{S}_{\text{ABDLOP}}$ as the last hybrid in a sequence of three simulators $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_{\text{ABDLOP}}$, the first two of which are allowed to use the secret $(\mathbf{s}_1, \mathbf{m})$ and are only a proof device, while the final $\mathcal{S}_{\text{ABDLOP}}$ uses none of it.

Protocol 1 Hybrid $\mathcal{S}_0$

Simulator: $\mathcal{S}_0$

Input: $\mathbf{s}_1, \mathbf{m}$

- 1: Sample $c \leftarrow \mathcal{C}$
 - 2: Sample $\mathbf{s}_2 \leftarrow \chi^{m_2}$
 - 3: Generate $(\mathbf{t}_A, \mathbf{t}_B)$ honestly under fresh randomness.
 - 4: Sample $\mathbf{z}_1 \leftarrow D_{\mathbf{s}_1}^{m_1}$ ▷ Sample masked responses with rejection
 - 5: Sample $\mathbf{z}_2 \leftarrow D_{\mathbf{s}_2}^{m_2}$ conditioned on $\langle \mathbf{s}_2, \mathbf{z}_2 \rangle \geq 0$ ▷ i.e. run the Rej_2 branch of [LNP22, Lemma 2.14].
 - 6: $\mathbf{w} := A_1 \mathbf{z}_1 + A_2 \mathbf{z}_2 - c \mathbf{t}_A$ ▷ Recompute the commitment openings
 - 7: $\mathbf{v} := R_1 \mathbf{z}_1 + R_m(c \mathbf{t}_B - B \mathbf{z}_2) - c \mathbf{u}$
 - 8: **return** $((\mathbf{t}_A, \mathbf{t}_B), (\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2))$
-

By [LNP22, Lemma 2.14] (the Rej_2 case), the distribution output by $\mathcal{S}_0$ is statistically close to that of an actual non-aborting execution between the honest prover and the honest verifier.

Protocol 2 Hybrid $\mathcal{S}_1$

Simulator: $\mathcal{S}_1$

Input: $\mathbf{s}_1, \mathbf{m}$

- 1: Sample $c \leftarrow \mathcal{C}$
 - 2: Sample $\mathbf{s}_2 \leftarrow \chi^{m_2}$
 - 3: Sample $\mathbf{u}' \leftarrow \mathbb{R}_{\mathbb{F}}^{n+\ell}$
 - 4: Set $\begin{pmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{pmatrix} := \mathbf{u}' + \begin{pmatrix} A_1 \mathbf{s}_1 \\ \mathbf{m} \end{pmatrix}$ ▷ Replace the honest commitment by a uniformly random one
 - 5: Sample $\mathbf{z}_1 \leftarrow D_{\mathbf{s}_1}^{m_1}$ ▷ Sample masked responses with rejection
 - 6: Sample $\mathbf{z}_2 \leftarrow D_{\mathbf{s}_2}^{m_2}$ conditioned on $\langle \mathbf{s}_2, \mathbf{z}_2 \rangle \geq 0$ ▷ i.e. run the Rej_2 branch of [LNP22, Lemma 2.14]
 - 7: $\mathbf{w} := A_1 \mathbf{z}_1 + A_2 \mathbf{z}_2 - c \mathbf{t}_A$ ▷ Recompute the commitment openings
 - 8: $\mathbf{v} := R_1 \mathbf{z}_1 + R_m(c \mathbf{t}_B - B \mathbf{z}_2) - c \mathbf{u}$
 - 9: **return** $((\mathbf{t}_A, \mathbf{t}_B), (\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2))$
-

Note that $\mathcal{S}_1$ runs identically to $\mathcal{S}_0$, except that instead of generating $(\mathbf{t}_A, \mathbf{t}_B)$ honestly, it samples $\mathbf{u}' \leftarrow \mathbb{R}_{\mathbb{F}}^{n+\ell}$. Now, we claim that if a PPT adversary $\mathcal{A}$ distinguishes the outputs of $\mathcal{S}_0$ and $\mathcal{S}_1$ with advantage ε , there is a PPT $\mathcal{B}$ solving Extended-MLWE $_{n+\ell, 1, m_2-n-\ell, \chi, \mathcal{C}, \mathbf{s}_2}$ with advantage at least $\varepsilon/2$. Given an Extended-MLWE tuple $(\mathbf{C}, \mathbf{u}^*, \mathbf{z}_2, \mathbf{b})$ for $\mathbf{C} := \begin{pmatrix} A_2 \\ B \end{pmatrix}$, $\mathcal{B}$ sets $\begin{pmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{pmatrix} := \mathbf{u}^* + \begin{pmatrix} A_1 \mathbf{s}_1 \\ \mathbf{m} \end{pmatrix}$ and simulates the rest of the transcript identically to $\mathcal{S}_0/\mathcal{S}_1$, then outputs the result to $\mathcal{A}$. Condition on $\mathbf{b} = 1$ (which happens with probability $\geq 1/2$): if $\mathbf{u}^* = \mathbf{C} \mathbf{s}_2$, the output of $\mathcal{B}$ has exactly the distribution of $\mathcal{S}_0$'s output; if $\mathbf{u}^*$ is uniform, it has exactly the distribution of $\mathcal{S}_1$'s. Hence $\mathcal{B}$'s advantage, conditioned on $\mathbf{b} = 1$, matches $\mathcal{A}$'s, giving unconditional advantage at least $\varepsilon/2$.

Protocol 3 Final Simulator $\mathcal{S}_{\text{ABDLOP}}$

Simulator: $\mathcal{S}_{\text{ABDLOP}}$

Input: public parameters and no secret information.

- 1: Sample $c \leftarrow \mathcal{C}$
- 2: Sample $\mathbf{s}_2 \leftarrow \chi^{m_2}$
- 3: Sample $(\mathbf{t}_A, \mathbf{t}_B) \leftarrow \mathbb{R}_{\mathbb{F}}^{n+\ell}$ ▷ Sample the commitment directly, with no reference to $\mathbf{s}_1, \mathbf{m}$
- 4: Sample $\mathbf{z}_1 \leftarrow D_{\mathbf{s}_1}^{m_1}$ ▷ Sample masked responses with rejection

5: Sample $\mathbf{z}_2 \leftarrow D_{\mathbf{s}_2}^{m_2}$ conditioned on $\langle \mathbf{s}_2, \mathbf{z}_2 \rangle \geq 0$ $\triangleright$ i.e. run the Rej_2 branch of [LNP22, Lemma 2.14]
6: $\mathbf{w} := A_1 \mathbf{z}_1 + A_2 \mathbf{z}_2 - c \mathbf{t}_A$ $\triangleright$ Recompute the commitment openings
7: $\mathbf{v} := R_1 \mathbf{z}_1 + R_m(c \mathbf{t}_B - B \mathbf{z}_2) - c \mathbf{u}$
8: **return** $((\mathbf{t}_A, \mathbf{t}_B), (\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2))$ $\triangleright$ Identical distribution to $\mathcal{S}_1$, since $\mathbf{u}'$ there is uniform over $\mathbb{R}_{\mathbb{F}}^{n+\ell}$

Combining the two hybrid steps ($\mathcal{S}_0 \approx_c \mathcal{S}_1$ under Extended-MLWE $_{n+\ell, 1, m_2-n-\ell, \chi, c, \mathbf{s}_2}$, and $\mathcal{S}_1 \equiv \mathcal{S}_{\text{ABDLOP}}$ as distributions) together with $\mathcal{S}_0 \approx_s$ (real transcript) from [LNP22, Lemma 2.4], the triangle inequality gives that $\mathcal{S}_{\text{ABDLOP}}$'s output is computationally indistinguishable from a real transcript between the honest prover and honest (possibly malicious-but-passive) verifier. $\blacksquare$

In order to prove the quadratic-relations, we need to employ the proof system of [LNP22, §4, Fig. 6]; $\Pi_{\text{ABDLOP}}^{(2)}((s_2, s_1, m), \sigma, f)$. For our use in Protocol 6 we instantiate with $k = 1$, $\sigma = \text{id}$ (so $s = (s_1, m)$ with no automorphism-lifting), since we only ever need to certify full-ring identities such as $m_{sq,i} - y'_{i,1} \cdot y'_{i,1} = 0$, not single coefficients thereof.

Lemma 3.7 ([LNP22, Theorem 4.2, Lemma 4.3]). *Protocol $\Pi_{\text{ABDLOP}}^{(2)}((s_2, s_1, m), \sigma, f)$ is complete, and there is an extractor $\mathcal{E}$ which, given rewindable black-box access to a prover convincing the verifier with non-negligible probability, outputs $(\bar{s}_1, \bar{s}_2, \bar{m}) \in \mathbb{R}_{\mathbb{F}}^{m_1+m_2+\ell}$ and $\bar{c} \in \bar{\mathcal{C}}$ such that*

$$\begin{pmatrix} t_A \\ t_B \end{pmatrix} = \begin{pmatrix} A_1 \\ 0 \end{pmatrix} \bar{s}_1 + \begin{pmatrix} A_2 \\ B \end{pmatrix} \bar{s}_2 + \begin{pmatrix} 0 \\ \bar{m} \end{pmatrix}, \quad f_j(\bar{s}_1, \bar{m}) = 0 \text{ for all } j \in [N],$$

together with the norm bounds $\|\bar{c} \bar{s}_i\| \leq 2\mathbf{s}_i \sqrt{2m_i d}$ ($i = 1, 2$), or else a solution to $\text{MSIS}_{n, m_1+m_2, B}$ for $[A_1 \mid A_2]$. Similarly to Lemma 3.6, there exists a simulator $\mathcal{S}_{\text{ABDLOP}}^{\text{square}}$.

To simultaneously prove N quadratic relations $f_1, \dots, f_N$ (which is what we need below, in order to batch the square/cube/product identities into a single sub-protocol call and a single round), the verifier first sends batching challenges $\mu_1, \dots, \mu_N \leftarrow \mathbb{R}_{\mathbb{K}}$ and the parties run $\Pi_{\text{ABDLOP}}^{(2)}((s_2, s_1, m), \sigma, \sum_{j=1}^N \mu_j f_j)$; we denote this $\Pi_{\text{ABDLOP, many}}^{(2)}((s_2, s_1, m), \sigma, (f_1, \dots, f_N))$, cf. [LNP22, Fig. 7]. This adds only a single round of N challenge ring elements over the basic protocol $\Pi_{\text{ABDLOP}}^{(2)}((s_2, s_1, m), \sigma, f)$, and its soundness error degrades additively by at most $q^{-d/2}$ (cf. also [LNP22, §4.3]). More precisely, at $q \equiv 2l + 1 \pmod{4l}$ with $l = 2$ the polynomial $X^d + 1$ has $\varphi(2d)/\text{ord}_{2d}(q) = 2$ irreducible factors over $\mathbb{F}$, and over the quadratic extension $\mathbb{K}$ the order halves, giving 4 factors of degree $d/4$. By the Chinese Remainder Theorem $\mathbb{R}_{\mathbb{K}} \cong \prod_{i=1}^4 \mathbb{K}[X]/(f_i)$ with each quotient a field of size $|\mathbb{K}|^{d/4} = q^{d/2}$. A uniform $\mu \in \mathbb{R}_{\mathbb{K}}$ therefore annihilates a fixed nonzero element with probability at most $q^{-d/2}$ (from the random linear combination collapsing to zero on a nonzero vector of relations), which is negligible for our parameter choices.

A lemma analogous to Lemma 3.7 holds true for the batched version $\Pi_{\text{ABDLOP, many}}^{(2)}$ for N relations.

3.2.1. Vanishing constant coefficients. Several steps of our reduction do not prove a full-ring identity, but only that the constant coefficient of a committed quantity vanishes.

Protocol $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ proves $\mathbf{u} = \mathbf{R}_1 \mathbf{s}_1 + \mathbf{R}_m \mathbf{m}$ for a public $\mathbf{u} \in \mathbb{R}_{\mathbb{K}}^N$, i.e. a full-ring identity. It cannot be invoked directly on a statement about the constant term, since the honest prover's function $f(\mathbf{s}_1, \mathbf{m})$ is not the zero ring element and the verifier cannot compute its non-constant coefficients. We therefore recall the reduction of [LNP22, §3.2 (Figure 5), §4.3 (Figure 8)] from a family of constant-term-statements to a single full-ring statement, in the $\mathbb{K}$ -valued form justified in §3.3.5.

Protocol 4 $\Pi_{\text{many}}^{(\text{ct})}((s_2, s_1, \mathbf{m}), (f_1, \dots, f_N))$ – proof of $\text{ct}(f_j(\mathbf{s}_1, \mathbf{m})) = 0$ for all $j \in [N]$, for $\mathbb{R}_{\mathbb{K}}$ -linear (resp. quadratic) f_j .

Private information: $(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2)$ as in Protocol $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$.

Public information: $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}$, the commitment $(\mathbf{t}_A, \mathbf{t}_B)$, and the functions $f_1, \dots, f_N$.

- 1: $\mathcal{P}$: sample the *garbage term* $g \xleftarrow{\$} \{x \in \mathbb{R}_{\mathbb{K}} : \text{ct}(x) = 0\}$, append it to the committed message vector $\mathbf{m}$ (so $\ell \mapsto \ell + 1$), and send the resulting commitment to $\mathcal{V}$. $\triangleright$ One extra BDLOP message slot; g must be fixed *before* Step 2.
- 2: $\mathcal{V} \rightarrow \mathcal{P}$: sample $\vec{\gamma} = (\gamma_1, \dots, \gamma_N) \xleftarrow{\$} \mathbb{K}^N$ and send it.

3: $\mathcal{P} \rightarrow \mathcal{V}$: compute and send, *in the clear*,

$$h := \sum_{j=1}^N \gamma_j f_j(\mathbf{s}_1, \mathbf{m}) + g \in \mathbb{R}_{\mathbb{K}}.$$

4: $\mathcal{V}$: check $\text{ct}(h) = 0$; abort otherwise.

5: $\mathcal{P}, \mathcal{V}$: run $\Pi_{\text{many}}^{(1)}$ on the *single* full-ring linear relation

$$\sum_{j=1}^N \gamma_j f_j(\mathbf{s}_1, \mathbf{m}) + g = h,$$

i.e. with $N = 1$, public $\mathbf{u} := h$, and $\mathbf{R}_1, \mathbf{R}_m$ assembled from $\vec{\gamma}$ and the f_j (the g -column of $\mathbf{R}_m$ being the identity).

▷ If the f_j are quadratic, run $\Pi_{\text{ABDLOP}, \text{many}}^{(2)}$ on the single quadratic relation $\sum_j \gamma_j f_j + g - h = 0$ instead; nothing else changes.

Lemma 3.8. *Protocol 4 is complete, and, under the hypotheses of Lemma 3.6 (resp. Lemma 3.7), is a proof of knowledge of an opening $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}})$ of $(\mathbf{t}_A, \mathbf{t}_B)$ satisfying $\text{ct}(f_j(\bar{\mathbf{s}}_1, \bar{\mathbf{m}})) = 0$ for every $j \in [N]$, with knowledge-soundness error exceeding that of $\Pi_{\text{many}}^{(1)}$ (resp. $\Pi_{\text{ABDLOP}, \text{many}}^{(2)}$) by an additive $1/|\mathbb{K}|$. Moreover it is honest-verifier zero-knowledge. More precisely, h is distributed uniformly on $\{x \in \mathbb{R}_{\mathbb{K}} : \text{ct}(x) = 0\}$, independently of $(\mathbf{s}_1, \mathbf{m})$, so the simulator samples h from that set and invokes $\mathcal{S}_{\text{ABDLOP}}$ on the remaining transcript.*

Proof. Completeness: is immediate: $\text{ct}(h) = \sum_j \gamma_j \text{ct}(f_j) + \text{ct}(g) = 0$ for an honest prover, and the Step 5 relation holds in $\mathbb{R}_{\mathbb{K}}$ by construction.

Knowledge soundness: The extractor of Lemma 3.6 applied at Step 5 returns an opening with $\sum_j \gamma_j f_j(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}) + \bar{g} = h$ in $\mathbb{R}_{\mathbb{K}}$. Taking constant coefficients and using $\mathcal{V}$'s check, $\sum_j \gamma_j \text{ct}(f_j) + \text{ct}(\bar{g}) = 0$. Both $\bar{g}$ and the f_j 's arguments are fixed by the commitment of Step 1, hence independent of $\vec{\gamma}$. In particular, we do not need $\text{ct}(\bar{g}) = 0$, which $\mathcal{V}$ cannot check. Suppose $\text{ct}(f_{j_0}) \neq 0$ for some j_0 . Conditioning on $\{\gamma_j\}_{j \neq j_0}$ and on $\text{ct}(\bar{g})$, the left-hand side is a nonconstant affine function of $\gamma_{j_0} \in \mathbb{K}$, so it vanishes with probability exactly $1/|\mathbb{K}|$. (This is the $\mathbb{K}$ -valued masking step analysed componentwise in §3.3.5, whose determinant computation gives the same $1/|\mathbb{F}|^2 = 1/|\mathbb{K}|$.)

Zero-knowledge: Conditioned on $\vec{\gamma}$ and on $(\mathbf{s}_1, \mathbf{m})$, the map $g \mapsto \sum_j \gamma_j f_j + g$ is a bijection of $\{x : \text{ct}(x) = 0\}$ onto $\{x : \text{ct}(x) = \sum_j \gamma_j \text{ct}(f_j)\} = \{x : \text{ct}(x) = 0\}$, so h is uniform on that set. ■

Remark 3.9. *Over $\mathbb{Z}_q$ the masking step of [LNP22, Figure 5] has soundness error $1/q$ and must be repeated $\lceil \lambda / \log_2 q \rceil$ times. Because our challenges are drawn from $\mathbb{K} = \mathbb{F}_{q^2}$ with $|\mathbb{K}| \approx 2^{128}$, a single execution already gives $1/|\mathbb{K}| \approx 2^{-128} \leq 2^{-\lambda}$ and thus no repetition is needed. The overhead of Protocol 4 over $\Pi_{\text{many}}^{(1)}$ is one committed ring element, one challenge round, and one ring element h sent in the clear.*

3.2.1.1. Anchoring salt aggregates to committed messages. In Protocol 6 we must anchor the transmitted Sum-Check claimed sum $\sigma := \zeta P + T$ (Step 7) and the verifier-computed target $u^* := h - \text{eq}(r', \alpha) \cdot (c_F + \delta_0 \cdot c_N) - \delta_1 \cdot c_E$ (Step 18) to the commitments already in $\mathcal{V}$'s view, so that a cheating prover cannot supply an arbitrary field value. Note that u^* is never transmitted as it is computed by $\mathcal{V}$ alone, and h is public (it is the Sum-Check's last-round output).

The first of these is directly a constant-coefficient relation in the committed messages coeffs and $\{y_{i,j}\}$ with public target σ , and needs nothing beyond Protocol 4 on a concatenation of the commitments (Definition 3.15 below). The second is not, and has instead the shape

$$(3.1) \quad u^* \stackrel{?}{=} \text{ct}\left(\sum_{n=1}^{N_c} w_n \cdot (B \mathbf{s}_2^{(n)})_{\iota(n)}\right), \quad w_n \in \mathbb{R}_{\mathbb{K}}, \iota(n) \in [\ell] \text{ public},$$

a constant-coefficient condition on a public weighted combination of the *BDLOP salt aggregates* of previously issued commitments. (Each w_n is a packaged rotation $\pi_{i,j}$ from Lemma 3.12 times a challenge product such as $\gamma_i^{(1)}$, $\zeta \delta_1$, or $\text{eq}(r', \alpha)$; $\iota(n)$ records which row of B the weight acts on, a datum suppressed in the informal statement of Equation (3.1) but needed below.)

At first sight Equation (3.1) is not of the form Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ certifies, which is a relation in $(\mathbf{s}_1, \mathbf{m})$ and not in $\mathbf{s}_2$. It becomes so after one substitution, and no relabelling of the two slots is required.

Lemma 3.10 (Salt/message substitution). *Let $\{(\mathbf{t}_A^{(n)}, \mathbf{t}_B^{(n)})\}_{n \in [N_c]}$ be ABDLOP commitments under shared public matrices (A_1, A_2, B) , each with a nonempty BDLOP part, so that $\mathbf{t}_B^{(n)} = B\mathbf{s}_2^{(n)} + \mathbf{m}^{(n)}$. Then for any public $w_n \in \mathbb{R}_{\mathbb{K}}$ and $\iota(n) \in [\ell]$, Equation (3.1) holds if and only if*

$$\text{ct}\left(\sum_{n=1}^{N_c} w_n \mathbf{m}_{\iota(n)}^{(n)} - \mathbf{u}\right) = 0, \quad \mathbf{u} := \sum_{n=1}^{N_c} w_n (\mathbf{t}_B^{(n)})_{\iota(n)} - u^* \cdot 1 \in \mathbb{R}_{\mathbb{K}},$$

and $\mathbf{u}$ is computable by $\mathcal{V}$ from the transmitted commitments and the public weights alone.

Proof. Substituting $(B\mathbf{s}_2^{(n)})_{\iota(n)} = (\mathbf{t}_B^{(n)})_{\iota(n)} - \mathbf{m}_{\iota(n)}^{(n)}$ into Equation (3.1) and using $\mathbb{K}$ -linearity of ct gives $\text{ct}(\sum_n w_n (\mathbf{t}_B^{(n)})_{\iota(n)}) - \text{ct}(\sum_n w_n \mathbf{m}_{\iota(n)}^{(n)}) = u^*$, which rearranges to the displayed identity. Computability of $\mathbf{u}$ is immediate, as every $\mathbf{t}_B^{(n)}$ has been transmitted. $\blacksquare$

Remark 3.11. *The above represents the action $\mathcal{V}$ already performs at Step 16 of Protocol 6, where c_F, c_N, c_E are derived from the BDLOP components $\text{Commit}(\cdot)_2 = B\mathbf{s}_2 + m$ and recorded as $c_F = F(\vec{r}') + \text{ct}(B \cdot \mathbf{s}_{2,F})$. Every salt aggregate the protocol ever needs is visible to $\mathcal{V}$ only in the form (public BDLOP component) – (committed message), which is exactly the form Lemma 3.10 exploits. The hypothesis $\ell^{(n)} \geq 1$ is essential and is not automatic: were a block committed with $\ell^{(n)} = 0$, its $B\mathbf{s}_2^{(n)}$ would appear nowhere in the transcript and $\mathbf{u}$ would not be computable. Our standing convention (§2.1) fixes $\ell^{(n)} \geq 1$ throughout.*

3.2.1.2. Coefficient extraction as a ring multiplication.

Lemma 3.12 (Rotation extraction). *Let $\tau_1 := 1$ and, for $\ell \in \{2, \dots, d\}$, let $\tau_\ell := -X^{d-\ell+1} \in \mathbb{R}_{\mathbb{K}}$. Then for every $a \in \mathbb{R}_{\mathbb{K}}$,*

$$\text{ct}(\tau_\ell \cdot a) = \text{cf}(a)_\ell.$$

Proof. Write $a = \sum_{i=0}^{d-1} a_i X^i$. Since $X^d \equiv -1 \pmod{\Phi(X)}$, multiplying by $X^{d-\ell+1}$ sends the coefficient of $X^{\ell-1}$ to the constant slot with a sign flip from the single wrap-around; the extra -1 folded into τ_ℓ exactly cancels it, and no other coefficient of a contributes to $\text{ct}(\tau_\ell a)$ because every other exponent $i + d - \ell + 1$ reduces to a nonzero residue mod d (or, for $\ell = 1$, no wrap occurs at all). $\blacksquare$

More generally, for a sum $\sum_{\ell=1}^d w_\ell \text{cf}(a)_\ell$ with public weights $w_\ell \in \mathbb{K}$, we define the packaged rotation as

$$\rho := \sum_{\ell=1}^d w_\ell \tau_\ell \in \mathbb{R}_{\mathbb{K}}, \quad \text{so that} \quad \sum_{\ell=1}^d w_\ell \text{cf}(a)_\ell = \text{ct}(\rho \cdot a).$$

This collapses an entire inner sum over $\ell \in [d]$ (as appears in, e.g., T , and $S_{2,E}$) into a single ring multiplication by a scalar computable by both parties from the public challenges $\gamma, \zeta, \delta_0, \delta_1$.

Lemma 3.13 (Batched degree-0 check). *Fix $a \in \mathbb{R}_{\mathbb{K}}$. For $\{\mu_\ell\}_{\ell=2}^d \xleftarrow{\$} \mathbb{K}^{d-1}$, let $\pi_\mu := \sum_{\ell=2}^d \mu_\ell \tau_\ell \in \mathbb{R}_{\mathbb{K}}$ be the packaged rotation of the weights $(0, \mu_2, \dots, \mu_d)$. If a is not of degree 0, i.e. $\text{cf}(a)_\ell \neq 0$ for some $\ell \geq 2$, then*

$$\Pr_{\{\mu_\ell\}} [\text{ct}(\pi_\mu \cdot a) = 0] \leq \frac{1}{|\mathbb{K}|}.$$

Consequently, for a fixed finite family $\{a_i\}_{i \in [N]} \subset \mathbb{R}_{\mathbb{K}}$, sampling a single $\{\mu_\ell\}_{\ell=2}^d$ together with an aggregation vector $\vec{\gamma} \xleftarrow{\$} \mathbb{K}^N$ and checking the single condition $\text{ct}(\sum_{i \in [N]} \gamma_i \pi_\mu \cdot a_i) = 0$ certifies that every a_i is degree-0, except with probability at most $2/|\mathbb{K}|$.

Proof. By Lemma 3.12, $\text{ct}(\pi_\mu \cdot a) = \sum_{\ell=2}^d \mu_\ell \text{cf}(a)_\ell$, a nonzero $\mathbb{K}$ -linear form in $(\mu_2, \dots, \mu_d)$ whenever some $\text{cf}(a)_\ell \neq 0$ for $\ell \geq 2$. A nonzero linear form over $\mathbb{K}$ vanishes at a uniformly random point with probability exactly $1/|\mathbb{K}|$, giving the first claim. For the aggregated statement, suppose a_{i_0} is not degree-0 for some i_0 . By the first claim, $A_{i_0} := \text{ct}(\pi_\mu a_{i_0}) \neq 0$ except with probability $1/|\mathbb{K}|$ over $\{\mu_\ell\}$. Conditioned on that and on $\{\gamma_i\}_{i \neq i_0}$, the quantity $\sum_i \gamma_i A_i$ is a nonconstant affine function of γ_{i_0} and so vanishes with probability $1/|\mathbb{K}|$.

The claim follows, up to probability $2/|\mathbb{K}|$, as aggregating with independent γ_i requires only one violated a_i to survive, not all of them. $\blacksquare$

Remark 3.14. Lemma 3.13 is what lets a single batched ABDLOP linear proof of knowledge certify that a committed ring element is a bona fide scalar (up to soundness error $2/|\mathbb{K}|$ in the number N of elements batched).

3.2.1.3. The concatenated instance. Suppose the prover has previously issued N_c ABDLOP commitments $\{(\mathbf{t}_A^{(n)}, \mathbf{t}_B^{(n)})\}_{n \in [N_c]}$ with known openings $(\mathbf{s}_1^{(n)}, \mathbf{s}_2^{(n)}, \mathbf{m}^{(n)})$ under the shared matrices (A_1, A_2, B) .

Definition 3.15. Define the concatenated ABDLOP instance

$$\begin{aligned} \mathbf{s}_1^{\text{cat}} &:= (\mathbf{s}_1^{(1)}, \dots, \mathbf{s}_1^{(N_c)}) \in \mathbb{R}_{\mathbb{F}}^{N_c m_1}, & \mathbf{s}_2^{\text{cat}} &:= (\mathbf{s}_2^{(1)}, \dots, \mathbf{s}_2^{(N_c)}) \in \mathbb{R}_{\mathbb{F}}^{N_c m_2}, \\ \mathbf{m}^{\text{cat}} &:= (\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(N_c)}, g), & A_i^{\text{cat}} &:= \text{blockdiag}(A_i, \dots, A_i) \ (i = 1, 2), & B^{\text{cat}} &:= \text{blockdiag}(B, \dots, B), \\ \mathbf{t}_A^{\text{cat}} &:= (\mathbf{t}_A^{(n)})_n, & \mathbf{t}_B^{\text{cat}} &:= (\mathbf{t}_B^{(n)})_n, & \mathbf{R}_1 &:= 0, & \mathbf{R}_m &:= (w_1 e_{\iota(1)}^\top \mid \dots \mid w_{N_c} e_{\iota(N_c)}^\top), \\ & & f^{\text{anc}}(\mathbf{s}_1^{\text{cat}}, \mathbf{m}^{\text{cat}}) &:= \mathbf{R}_m \mathbf{m}^{\text{cat}} - \mathbf{u} \in \mathbb{R}_{\mathbb{K}}, \end{aligned}$$

where g is the garbage term of Protocol 4, e_ι is the ι -th standard basis vector of $\mathbb{R}_{\mathbb{K}}^\ell$, and $\mathbf{u} \in \mathbb{R}_{\mathbb{K}}$ is a public target. We write $\Pi^{\text{anc}}(\{(w_n, \iota(n))\}_n, \mathbf{u})$ for the execution of $\Pi_{\text{many}}^{(\text{ct})}$ (Protocol 4) with $N = 1$ on the single relation $\text{ct}(f^{\text{anc}}) = 0$, instantiated with $(A_1^{\text{cat}}, A_2^{\text{cat}}, B^{\text{cat}}, \mathbf{R}_1, \mathbf{R}_m, \mathbf{u})$ and rejection-sampling widths

$$\mathbf{s}_1^{\text{cat}} := \gamma_1 \eta \alpha \sqrt{N_c m_1 d}, \quad \mathbf{s}_2^{\text{cat}} := \gamma_2 \eta \nu \sqrt{N_c m_2 d}.$$

The two calls of Protocol 6 differ only in how $\mathbf{u}$ is obtained: at Step 7 it is the transmitted scalar σ , and at Step 18 it is assembled by $\mathcal{V}$ from the $\mathbf{t}_B^{(n)}$ as in Lemma 3.10.

By construction $A_1^{\text{cat}} \mathbf{s}_1^{\text{cat}} + A_2^{\text{cat}} \mathbf{s}_2^{\text{cat}} = (A_1 \mathbf{s}_1^{(n)} + A_2 \mathbf{s}_2^{(n)})_n = \mathbf{t}_A^{\text{cat}}$ and $B^{\text{cat}} \mathbf{s}_2^{\text{cat}} + \mathbf{m}^{\text{cat}} = \mathbf{t}_B^{\text{cat}}$ (up to the one extra BDLOP slot carrying g), so no commitment is recomputed: the parties merely repackaging openings they already hold. The widths are the ordinary widths of $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ under $m_1 \mapsto N_c m_1$, $m_2 \mapsto N_c m_2$, with the ℓ_∞ -to- ℓ_2 conversion of §2.5.2 applied to each of the N_c blocks, as [LNP22, §2.6] requires.

Lemma 3.16. Let $B^{\text{anc}} := 8\eta \sqrt{(\mathbf{s}_1^{\text{cat}} \sqrt{2N_c m_1 d})^2 + (\mathbf{s}_2^{\text{cat}} \sqrt{2N_c m_2 d})^2}$ and assume $\text{MSIS}_{n, m_1 + m_2, B^{\text{anc}}}$ is hard.

Then Π^{anc} is complete, and is a proof of knowledge of openings $(\bar{\mathbf{s}}_1^{(n)}, \bar{\mathbf{s}}_2^{(n)}, \bar{\mathbf{m}}^{(n)})_{n \in [N_c]}$ of the previously fixed commitments $\{(\mathbf{t}_A^{(n)}, \mathbf{t}_B^{(n)})\}$ satisfying $\text{ct}(\sum_n w_n \bar{\mathbf{m}}_{\iota(n)}^{(n)} - \mathbf{u}) = 0$ (equivalently, when $\mathbf{u}$ is taken as in Lemma 3.10, Equation (3.1) on the extracted salts) with knowledge-soundness error exceeding that of Lemma 3.6 by the additive $1/|\mathbb{K}|$ of Lemma 3.8. It is honest-verifier zero-knowledge, with simulator the composition of Lemma 3.8's with $\mathcal{S}_{\text{ABDLOP}}$, under $\text{Extended-MLWE}_{n+\ell+1, N_c, m_2-n-\ell-1, \chi, C, \mathbf{s}_2}$.

Proof. Completeness. An honest prover satisfies $\text{ct}(f^{\text{anc}}) = 0$, directly at Step 7 and by Lemma 3.10 at Step 18; the norm hypotheses $\|\mathbf{s}_1^{\text{cat}}\| \leq \alpha \sqrt{N_c m_1 d}$ and $\|\mathbf{s}_2^{\text{cat}}\|_\infty \leq \nu$ hold blockwise, so the widths of Definition 3.15 are the ones Lemma 3.6 prescribes, and completeness follows from Lemma 3.8.

Knowledge soundness. Lemma 3.8 reduces to the extractor of Lemma 3.6 applied at Step 5 of Protocol 4, which returns $(\bar{\mathbf{s}}_1^{\text{cat}}, \bar{\mathbf{s}}_2^{\text{cat}}, \bar{c})$ with $A_1^{\text{cat}} \bar{\mathbf{s}}_1^{\text{cat}} + A_2^{\text{cat}} \bar{\mathbf{s}}_2^{\text{cat}} = \mathbf{t}_A^{\text{cat}}$ and

$$\mathbf{R}_1 \bar{\mathbf{s}}_1^{\text{cat}} + \mathbf{R}_m (\mathbf{t}_B^{\text{cat}} - B^{\text{cat}} \bar{\mathbf{s}}_2^{\text{cat}}) = \mathbf{u}.$$

Since $\mathbf{R}_1 = 0$, this reads $\sum_n w_n (\mathbf{t}_B^{(n)} - B \bar{\mathbf{s}}_2^{(n)})_{\iota(n)} = \mathbf{u}$, i.e. the asserted relation on the extracted messages $\bar{\mathbf{m}}^{(n)} := \mathbf{t}_B^{(n)} - B \bar{\mathbf{s}}_2^{(n)}$, and hence by Lemma 3.10 precisely Equation (3.1) on the extracted salts. The block structure enters only at the level of binding. More precisely, a second opening yields a nonzero $\mathbf{x} = (\mathbf{x}^{(n)})_n$ of norm at most B^{anc} with $[A_1^{\text{cat}} \mid A_2^{\text{cat}}] \mathbf{x} = 0$, and since the matrix is block diagonal this holds blockwise; picking any n_0 with $\mathbf{x}^{(n_0)} \neq 0$ gives a $\text{MSIS}_{n, m_1 + m_2, B^{\text{anc}}}$ solution for $[A_1 \mid A_2]$ itself. The additive $1/|\mathbb{K}|$ comes from Lemma 3.8.

Honest-verifier zero-knowledge. The transcript consists of $\mathbf{t}_g$, the challenge $\tilde{\gamma}$, the ring element h , and one $\Pi_{\text{many}}^{(1)}$ transcript. By Lemma 3.8 the element h is uniform on $\{x \in \mathbb{R}_{\mathbb{K}} : \text{ct}(x) = 0\}$ independently of the witness, and the remainder is simulated by $\mathcal{S}_{\text{ABDLOP}}$ (Protocol 3). The hybrid argument $\mathcal{S}_0 \rightarrow \mathcal{S}_1$ of Lemma 3.6 goes through verbatim, one block at a time. In fact, the pseudorandomness required is that of

$((\frac{A_2}{B})|B_g, \cdot s_2^{(n)})$ for $s_2^{(n)} \leftarrow \chi^{m_2}$, which is the assumption already made for each commitment individually, and the N_c salt blocks are independently sampled. We can then proceed by a standard hybrid simulator procedure over $n \in [N_c]$ steps. Therefore we lose a factor N_c and no new assumption is introduced. ■

The cost of one Π^{anc} call is exactly that of one call to Protocol $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ on vectors of length $N_c \cdot m_1$ and $N_c \cdot m_2$, wrapped in one $\Pi_{\text{many}}^{(\text{ct})}$ with $N = 1$, i.e. $\mathcal{O}(N_c \cdot (m_1 + m_2) \cdot d)$ field operations, and no fresh commitment is computed beyond the single garbage slot for g . Overall this incurs $\mathcal{O}(t \cdot (k + K)) \times$ a single ZKPoK.

3.3. Instantiating ABDLOP over $R_{\mathbb{K}}$ Componentwise. Throughout this work, the ABDLOP commitment scheme and the accompanying zero-knowledge machinery of [LNP22] are used over a quadratic extension ring $R_{\mathbb{K}} = \mathbb{K}[X]/(\Phi(X))$, which by restriction (6) of §1.3 we take throughout to be $\Phi(X) = X^d + 1$ with d a power of two, as in [LNP22]. This section specifies precisely how that instantiation is done: rather than building an ABDLOP commitment whose matrices and messages live natively over $R_{\mathbb{K}}$ (which would require the hardness of MLWE/MSIS over $R_{\mathbb{K}}$ itself), we identify $\mathbb{K} = \mathbb{F}[Y]/(Y^2 - \nu)$ for ν a quadratic non-residue in $\mathbb{F}$, and commit to the $\{1, Y\}$ -coordinates of $R_{\mathbb{K}}$ -elements directly, as ordinary $R_{\mathbb{F}}$ -elements, and translate every $R_{\mathbb{K}}$ -relation used in the protocol into an equivalent pair of $R_{\mathbb{F}}$ -relations *once*, at the level of public parameters, before invoking [LNP22]'s proof system unchanged.

Following this approach, the ABDLOP commitment matrices are then genuinely uniform over $R_{\mathbb{F}}$, and security rests only on standard $R_{\mathbb{F}}$ -MLWE/MSIS, the same assumption already used throughout the rest of the paper, with no additional or non-standard hardness assumption required.

3.3.1. The Rank-Doubling Embedding. Writing $\mathbb{K} = \mathbb{F}(\sqrt{\nu})$ for a fixed non-residue $\nu \in \mathbb{F}$, every element of $R_{\mathbb{K}}$ decomposes uniquely as $a + bY$ with $a, b \in R_{\mathbb{F}}$ and $Y^2 = \nu$; recall that $R_{\mathbb{K}}$ is a free module of rank 2 over $R_{\mathbb{F}}$, since $R_{\mathbb{K}} = R_{\mathbb{F}} \otimes_{\mathbb{F}} \mathbb{K}$. This gives an $\mathbb{F}$ -algebra isomorphism between $R_{\mathbb{K}}$ and its regular representation acting on $R_{\mathbb{F}}^2$,

$$(3.2) \quad \psi : R_{\mathbb{K}} \longrightarrow \text{Mat}_2(R_{\mathbb{F}}), \quad a + bY \longmapsto \begin{pmatrix} a & \nu b \\ b & a \end{pmatrix},$$

extended entry-wise to matrices, $\psi : R_{\mathbb{K}}^{m \times k} \rightarrow R_{\mathbb{F}}^{2m \times 2k}$, together with a compatible vector-side embedding $\varphi : R_{\mathbb{K}}^k \rightarrow R_{\mathbb{F}}^{2k}$ satisfying $\psi(A)\varphi(z) = \varphi(Az)$ for all $A \in R_{\mathbb{K}}^{m \times k}$, $z \in R_{\mathbb{K}}^k$. Both maps are injective ring/module homomorphisms.

Let us remark that we do not use this decomposition as an embedding of an already-instantiated $R_{\mathbb{K}}$ -scheme, but as the definition of the witness representation itself. More precisely, every (secret or public) $R_{\mathbb{K}}$ -element that appears in the protocol is represented, from the start, as a pair of $R_{\mathbb{F}}$ -elements (a, b) , and every commitment is an ordinary ABDLOP commitment to such pairs over $R_{\mathbb{F}}$ (cf. §2.5.2, with A_1, A_2, B sampled uniformly over $R_{\mathbb{F}}$, not confined to $\psi(R_{\mathbb{K}}^{m \times k})$).

By construction ψ (resp. φ) is a bijection over $R_{\mathbb{K}} \leftrightarrow R_{\mathbb{F}}^2$ (resp. $R_{\mathbb{K}}^k \leftrightarrow R_{\mathbb{F}}^{2k}$), we do not need any consistency check for the condition (a, b) satisfy $x = a + bY$. Any pair $(a, b) \in R_{\mathbb{F}}^2$ is, by definition, the coordinate representation of some unique $x \in R_{\mathbb{K}}$, and every $R_{\mathbb{K}}$ -relation the protocol needs to enforce is instead enforced directly as a relation on (a, b) , built once from ψ at parameter-generation time.

What still needs to be checked is that this translation preserves

- the relations the ABDLOP protocols need to prove (linear and quadratic),
- the norm bounds, and
- the soundness of $\mathbb{K}$ -valued challenge masking.

We address these in the order.

3.3.2. Linear Relations. For a public $R_{\mathbb{K}}$ -linear relation $A_{\mathbb{K}}x = t_{\mathbb{K}}$ with $A_{\mathbb{K}} = A_a + A_bY$, $t_{\mathbb{K}} = t_a + t_bY$, $x = a + bY$, expanding $(A_a + A_bY)(a + bY) = t_a + t_bY$ and using $Y^2 = \nu$ gives

$$(3.3) \quad A_a a + \nu A_b b = t_a, \quad A_b a + A_a b = t_b,$$

i.e. exactly $\psi(A_{\mathbb{K}}) \cdot (a; b) = (t_a; t_b)$. This is a pair of $R_{\mathbb{F}}$ -linear relations in the committed values (a, b) , of precisely the form $R_1 s_1 + R_m m = u$ that $\Pi_{\text{many}}^{(1)}$ ([LNP22, Figure 4/5]) already proves for arbitrary public R_1, R_m, u over $R_{\mathbb{F}}$. No modification to the proof system is needed, as one simply instantiates R_1 and R_m as $\begin{pmatrix} A_a & \nu A_b \\ A_b & A_a \end{pmatrix}$, and u as $\begin{pmatrix} t_a \\ t_b \end{pmatrix}$ in place of whatever relation the base protocol was proving.

3.3.3. Quadratic Relations and Automorphisms. We now carry out the component-wise expansion of a quadratic relation, since this case genuinely produces cross terms between a and b that the linear case does not.

Fix a public $n = k(m_1 + \ell)$ -variate quadratic function $f_{\mathbb{K}}$ over $\mathbb{R}_{\mathbb{K}}$ of the form used in [LNP22, Figures 6–10],

$$(3.4) \quad f_{\mathbb{K}}(x) = x^{\top} R_2 x + r_1^{\top} x + r_0, \quad R_2 \in \mathbb{R}_{\mathbb{K}}^{n \times n}, \quad r_1 \in \mathbb{R}_{\mathbb{K}}^n, \quad r_0 \in \mathbb{R}_{\mathbb{K}},$$

(for the sake of simplicity, we write n for the relevant vector dimension here, suppressing the automorphism-stacking index k for the moment). Write $R_2 = R_a + R_b Y$, $r_1 = r_{1a} + r_{1b} Y$, $r_0 = r_{0a} + r_{0b} Y$ with $R_a, R_b \in \mathbb{R}_{\mathbb{F}}^{n \times n}$, $r_{1a}, r_{1b} \in \mathbb{R}_{\mathbb{F}}^n$, $r_{0a}, r_{0b} \in \mathbb{R}_{\mathbb{F}}$, and $x = a + bY$ with $a, b \in \mathbb{R}_{\mathbb{F}}^n$.

3.3.3.1. Expanding the quadratic term. Since Y is central and $Y^2 = \nu$,

$$(3.5) \quad R_2 x = (R_a + R_b Y)(a + bY) = \underbrace{(R_a a + \nu R_b b)}_{=:p} + \underbrace{(R_a b + R_b a)}_{=:q} Y,$$

and hence

$$(3.6) \quad x^{\top} (R_2 x) = (a + bY)^{\top} (p + qY) = (a^{\top} p + \nu b^{\top} q) + (a^{\top} q + b^{\top} p) Y.$$

Substituting p, q back in and collecting terms, the $\{1, Y\}$ -coordinates of $x^{\top} R_2 x$ are

$$(3.7) \quad (x^{\top} R_2 x)_a = a^{\top} R_a a + \nu b^{\top} R_a b + \nu a^{\top} R_b b + \nu b^{\top} R_b a,$$

$$(3.8) \quad (x^{\top} R_2 x)_b = b^{\top} R_a a + a^{\top} R_b a + a^{\top} R_a b + \nu b^{\top} R_b b.$$

Each of Equations (3.7), (3.8) is a genuine $\mathbb{R}_{\mathbb{F}}$ -quadratic form in the stacked vector $s := (a; b) \in \mathbb{R}_{\mathbb{F}}^{2n}$: writing $s^{\top} \hat{R} s = a^{\top} \hat{R}_{11} a + a^{\top} \hat{R}_{12} b + b^{\top} \hat{R}_{21} a + b^{\top} \hat{R}_{22} b$ and matching coefficients against Equation (3.7) and Equation (3.8) respectively gives

$$(3.9) \quad \hat{R}^{(a)} = \begin{pmatrix} R_a & \nu R_b \\ \nu R_b & \nu R_a \end{pmatrix}, \quad \hat{R}^{(b)} = \begin{pmatrix} R_b & R_a \\ R_a & \nu R_b \end{pmatrix}.$$

By direct substitution, we can compute that $s^{\top} \hat{R}^{(a)} s$ reproduces Equation (3.7) term-for-term, and likewise for $\hat{R}^{(b)}$ and Equation (3.8). Note that the two matrices are not equal and neither is simply $\psi(R_2)$ from §3.3.1, since $\hat{R}^{(a)}$ and $\hat{R}^{(b)}$ encode a bilinear form's two $\{1, Y\}$ -coordinates, a different (though closely related) construction from ψ 's regular representation of ring multiplication.

3.3.3.2. Expanding the linear and constant terms. Similarly,

$$(3.10) \quad r_1^{\top} x = (r_{1a} + r_{1b} Y)^{\top} (a + bY) = (r_{1a}^{\top} a + \nu r_{1b}^{\top} b) + (r_{1a}^{\top} b + r_{1b}^{\top} a) Y,$$

giving linear parts $\hat{r}_1^{(a)} = (r_{1a}; \nu r_{1b})$ and $\hat{r}_1^{(b)} = (r_{1b}; r_{1a})$ in the stacked variable $s = (a; b)$; the constant term splits as $r_0 = r_{0a} + r_{0b} Y$ trivially.

3.3.3.3. The resulting pair of relations. Collecting the above, $f_{\mathbb{K}}(x) = 0$ over $\mathbb{R}_{\mathbb{K}}$ is equivalent to the simultaneous pair of $\mathbb{R}_{\mathbb{F}}$ -quadratic relations ($s = (a; b)$)

$$(3.11) \quad s^{\top} \hat{R}^{(a)} s + \hat{r}_1^{(a)\top} s + r_{0a} = 0, \quad s^{\top} \hat{R}^{(b)} s + \hat{r}_1^{(b)\top} s + r_{0b} = 0,$$

each of exactly the form $f(x) = x^{\top} R_2 x + r_1^{\top} x + r_0$ handled natively by $\Pi^{(2)}$, and jointly of the many quadratic equations form handled by $\Pi_{\text{many}}^{(2)}$ (Protocol $\Pi_{\text{many}}^{(1)}$ $((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$, [LNP22, Figure 6–7]). As in the linear case, this is a restatement of the substitution $Y^2 = \nu$. The two relations are built once, from $(R_a, R_b, r_{1a}, r_{1b}, r_{0a}, r_{0b})$, at parameter-generation time, and handed to [LNP22]'s unmodified quadratic proof system. The same expansion applies verbatim to the evaluation functions F_j of [LNP22, Figure 8] (which only need their constant coefficient proved zero, not the full relation) and to the *vanishing constant coefficient* bookkeeping of [LNP22, §§4.3–4.4], since those sections operate on quadratic functions of exactly this shape.

3.3.3.4. Automorphisms. For $\sigma_i \in \text{Aut}(\mathbb{R}_{\mathbb{F}})$ acting only on the ring variable X , we extend it to $\mathbb{R}_{\mathbb{K}}$ by fixing Y , that is, we set $\sigma_i(a + bY) = \sigma_i(a) + \sigma_i(b)Y$. In other words, σ_i commutes with the $\{1, Y\}$ -decomposition and acts component-wise on (a, b) . This property is independent of the specific quadratic function being expanded and thus applies uniformly to the sections above. More precisely, the automorphism-stacked quadratic functions of [LNP22, §§4.1–4.4] (built from $(\sigma^i(s_1))_{i \in [k]}$, $(\sigma^i(m))_{i \in [k]}$) expand component-wise exactly as above, with σ_i simply distributing across the stacking index and the $\{1, Y\}$ -decomposition independently. In particular the automorphism-stability machinery of [LNP22, §§4.1–4.4] (the choice of challenge set $\mathcal{C}_{\text{pch}}^{\sigma}$ fixed

under σ , and the trace-based packing trick of [LNP22, Lemma 4.7]) applies verbatim to a and b separately, with no cross term to track beyond the $\hat{R}^{(a)}, \hat{R}^{(b)}$ bookkeeping already derived.

3.3.4. Norms. Recall that the ABDLOP machinery works with the ℓ_2 norm, even though we bound the salt norms in terms of the ℓ_∞ one; the conversion is then obtained by $\|x\|_2 \leq \sqrt{|x|}\|x\|_\infty$, for x the length of x in terms of field elements.

We define $\|x\|$ for $x = a + bY \in \mathbb{R}_\mathbb{K}$ to be the coefficient-vector ℓ_2 norm of (a, b) in the $\{1, Y\}$ basis, i.e. $\|x\|^2 := \|a\|^2 + \|b\|^2$. Under this convention φ is an *exact* isometry, $\|\varphi(x)\| = \|x\|$, with no distortion constant, since $\{1, Y\}$ is by construction the basis in which φ is coordinate extraction.

We emphasize that this norm bound is *not* obtained by instantiating the quadratic-relation translation of §3.3.3 with $f_\mathbb{K}(x) := \sigma_\mathbb{K}(x) \cdot x$, the $\mathbb{K}$ -level automorphism identity that LNP22's exact- ℓ_2 -norm machinery (§5.2 of [LNP22]) uses over $\mathbb{R}_\mathbb{F}$. Extending σ_{-1} to $\mathbb{R}_\mathbb{K}$ by fixing Y and computing $\sigma_\mathbb{K}(x)x = (\sigma(a) + \sigma(b)Y)(a + bY) = (\sigma(a)a + \nu\sigma(b)b) + (\sigma(a)b + \sigma(b)a)Y$ shows that the $\mathbb{F}$ -part of $\sigma_\mathbb{K}(x)x$ is $\|a\|^2 + \nu\|b\|^2$, *not* $\|a\|^2 + \|b\|^2$ – the two differ by the factor ν on the b -term precisely because $\nu \neq 1$ (it is a non-residue). Plugging $x = a + bY$ directly into the $\mathbb{K}$ -automorphism identity therefore proves the wrong quantity and must be avoided.

Instead, $\|a\|^2$ and $\|b\|^2$ are each obtained by applying the identity $\sigma_{-1}(a)a = \|a\|^2$ (resp. for b) *directly* over $\mathbb{R}_\mathbb{F}$, using the base-ring automorphism $\sigma_{-1} \in \text{Aut}(\mathbb{R}_\mathbb{F})$ already available in [LNP22]; these are two independent, ordinary $\mathbb{R}_\mathbb{F}$ -quadratic-with-automorphism statements on a and b separately, not an instance of the $\mathbb{R}_\mathbb{K}$ -relation translation. Consequently any norm bound $\|x\| \leq \beta$ required by the protocol (approximate range proof and exact norm proof of [LNP22, §§5.1–5.2]) is proved as the single joint quadratic constraint $\|a\|^2 + \|b\|^2 \leq \beta^2$ on the committed (a, b) , provable by the same machinery as any other exact-norm statement in [LNP22, §5.2] with no adaptation beyond summing the two independent square-norm terms; we are only bounding a norm on a fixed, exact coordinate system, not moving an error distribution across a reduction.

3.3.5. Soundness of $\mathbb{K}$ -Valued Challenge Masking. The one place requiring a genuine argument, rather than routine translation, is the masking step used in [LNP22, §3.2 (Figure 5)] and [LNP22, §4.3 (Figure 8, and its soundness-boosted version)], where a challenge is used to force a committed value's constant coefficient to be zero with soundness error $1/q$. In both cases the challenge is drawn as a *scalar*, $\gamma \in \mathbb{Z}_q$ (resp. $\gamma_{i,j} \in \mathbb{Z}_q$ for the boosted version) and not a ring element; this is what makes the argument below apply and should not be confused with the ring-valued combination challenges $\mu_1, \dots, \mu_N \in R_q$ [LNP22, Figure 7], which amortize N separate quadratic equations into one via a completely different (degree-counting) soundness argument and are not examined here. In our component-wise instantiation, that amortization step needs no $\mathbb{K}$ -valued analogue at all: once every relation has already been expanded into $\mathbb{R}_\mathbb{F}$ -equations on (a, b) (§§3.3.2–3.3.3), amortizing many such equations into one is done exactly as in the unmodified base protocol, entirely over $\mathbb{R}_\mathbb{F}$, with no reference to $\mathbb{K}$.

We now check that replacing the scalar $\gamma \in \mathbb{Z}_q$ of the masking step with a scalar $\gamma \in \mathbb{K}$ (i.e. $\gamma = \gamma_a + \gamma_b Y$, $\gamma_a, \gamma_b \in \mathbb{F}$) delivers the intended $1/q^2$ soundness, for the corresponding $\mathbb{R}_\mathbb{K}$ -valued relation now represented component-wise. Write $f_\mathbb{K}(s) = f_a + f_b Y \in \mathbb{R}_\mathbb{K}$ for the committed evaluation (so $f_a, f_b \in \mathbb{R}_\mathbb{F}$), and $g = g_a + g_b Y \in \mathbb{R}_\mathbb{K}$ for the masking element, sampled so that $\tilde{g} = 0$ in $\mathbb{K}$ (or, equivalently, since $\{1, Y\}$ is an $\mathbb{F}$ -basis, so that $\tilde{g}_a = \tilde{g}_b = 0$ in $\mathbb{F}$). This is exactly two independent copies of the base-protocol masking distribution (each of g_a, g_b uniform outside its constant coefficient, zero at the constant coefficient); no new sampling procedure is required beyond doubling the existing one.

The masking identity $h = \gamma \cdot f_\mathbb{K}(s) + g$, with $\gamma_a, \gamma_b \in \mathbb{F} \subset \mathbb{R}_\mathbb{F}$ acting as scalars, holds as an identity of full $\mathbb{R}_\mathbb{F}$ -polynomials:

$$h_a = \gamma_a f_a + \nu \gamma_b f_b + g_a, \quad h_b = \gamma_a f_b + \gamma_b f_a + g_b.$$

Taking constant coefficients (well-defined since γ_a, γ_b are scalars, so $\widetilde{\gamma_a f_a} = \gamma_a \tilde{f}_a$ etc.) and using $\tilde{g}_a = \tilde{g}_b = 0$:

$$\tilde{h}_a = \gamma_a \tilde{f}_a + \nu \gamma_b \tilde{f}_b, \quad \tilde{h}_b = \gamma_a \tilde{f}_b + \gamma_b \tilde{f}_a.$$

In order to achieve soundness, we require that if $(\tilde{f}_a, \tilde{f}_b) \neq (0, 0)$ (i.e. $\widetilde{f_\mathbb{K}(s)} \neq 0$ in $\mathbb{K}$), the probability over $(\gamma_a, \gamma_b) \leftarrow \mathbb{F}^2$ that $\tilde{h}_a = \tilde{h}_b = 0$ is negligible. This holds iff $(\gamma_a, \gamma_b) \in \ker M$ for

$$M = \begin{pmatrix} \tilde{f}_a & \nu \tilde{f}_b \\ \tilde{f}_b & \tilde{f}_a \end{pmatrix}, \quad \det M = \tilde{f}_a^2 - \nu \tilde{f}_b^2 = N_{\mathbb{K}/\mathbb{F}}(\tilde{f}_a + \tilde{f}_b Y),$$

the norm form of $\mathbb{K}/\mathbb{F}$. Since the norm is non-degenerate, this value is nonzero whenever $(\tilde{f}_a, \tilde{f}_b) \neq (0, 0)$. If $\tilde{f}_b \neq 0$ and $\tilde{f}_a^2 = \nu \tilde{f}_b^2$, then $\nu = (\tilde{f}_a/\tilde{f}_b)^2$ is a square, contradicting that ν is a non-residue; if $\tilde{f}_b = 0, \tilde{f}_a \neq 0$, then $\det M = \tilde{f}_a^2 \neq 0$ trivially. Hence M is invertible over $\mathbb{F}$, $\ker M = \{(0, 0)\}$, and

$$\Pr_{(\gamma_a, \gamma_b) \leftarrow \mathbb{F}^2} [\tilde{h}_a = 0 \wedge \tilde{h}_b = 0] = \frac{1}{|\mathbb{F}|^2} = \frac{1}{q^2}.$$

This confirms the component-wise instantiation preserves soundness, for the constant-coefficient masking steps of [LNP22, Figure 5–Figure 8], given only that ν is a non-residue and q is prime.

Remark 3.17 (Extension degree $e > 2$). *Nothing is special to the quadratic extension $\mathbb{K} = \mathbb{F}(\sqrt{\nu})$ (degree $e = 2$), and more secure and non-interactive versions will require $e = 4$. For a general $\mathbb{K} = \mathbb{F}[Y]/(g)$ with g irreducible of degree e , replace the rank-doubling’s ambient by the regular representation $\mathbb{K} \rightarrow \text{Mat}_e(\mathbb{R}_{\mathbb{F}})$ in the basis $\{1, Y, \dots, Y^{e-1}\}$; every $\mathbb{R}_{\mathbb{K}}$ -element is then committed as the e -tuple of its coordinates, an $\mathbb{R}_{\mathbb{K}}$ -linear relation becomes e $\mathbb{R}_{\mathbb{F}}$ -linear ones, and a quadratic relation e $\mathbb{R}_{\mathbb{F}}$ -quadratic ones, by the same expansion as in §3.3.2–§3.3.3. The norm convention of §3.3.4 becomes $\|x\|^2 := \sum_{i=0}^{e-1} \|a_i\|^2$, proved as e independent $\mathbb{R}_{\mathbb{F}}$ -automorphism statements. The masking argument of §3.3.5 goes through verbatim: for $\gamma \in \mathbb{K}$ the matrix M is the regular representation of $\tilde{f} \in \mathbb{K}$, so $\det M = N_{\mathbb{K}/\mathbb{F}}(\tilde{f}) \neq 0$ whenever $\tilde{f} \neq 0$, giving soundness $1/q^e = 1/|\mathbb{K}|$ as before; irreducibility of g replaces the non-residuosity of ν .*

The cost is the message-slot occupancy. Each $\mathbb{R}_{\mathbb{K}}$ -element consumes e of the ℓ BDLOP slots rather than 2, so at a fixed ℓ the packed commitment counts scale by $e/2$, and with them $\sum N_c$ and the masked-openings row that dominates. Arithmetic over $\mathbb{K}$ likewise costs $\mathcal{O}(e^{\log_2 3})$ base-field multiplications under Karatsuba rather than $\mathcal{O}(3)$. We do not give a tuned $e = 4$ parameter set here; re-optimising ℓ upward is the obvious first move, since ℓ enters the commitment size linearly but N_c , and hence the dominant opening cost, inversely.

3.3.6. Complexity. We conclude by recording the cost of the component-wise instantiation.

On the *size* axis, a linear $\mathbb{R}_{\mathbb{K}}$ -relation expands into two $\mathbb{R}_{\mathbb{F}}$ -relations on the stacked vector $(a; b)$ of twice the length, so the number of proved rows, the committed vector length, and hence the transmitted opening all exactly double. On the *arithmetic* axis it does not double. By §3.3.2 the relation is applied through

$$(3.12) \quad \psi(A_{\mathbb{K}}) = \begin{pmatrix} A_a & \nu A_b \\ A_b & A_a \end{pmatrix},$$

a $2m \times 2k$ block matrix, so a matrix-vector product costs four $\mathbb{R}_{\mathbb{F}}$ products of the original shape, or three under Karatsuba; the block structure is precisely what the Karatsuba method exploits. A *quadratic* relation carries the same factor for the same reason: each $\mathbb{R}_{\mathbb{K}}$ -product $(a_1 + b_1 Y)(a_2 + b_2 Y)$ requires the four $\mathbb{R}_{\mathbb{F}}$ -products $a_1 a_2, b_1 b_2, a_1 b_2, a_2 b_1$, three under Karatsuba at the cost of one extra addition. So both sub-protocol families carry a factor 3 to 4 in prover and verifier work, and a factor 2 in communication.

3.4. Libra-style masking. Let $f(X_1, \dots, X_{\ell})$ be a multivariate polynomial with partial degrees $\vec{d} = (d_1, \dots, d_{\ell})$ and assume that we wish to prove that $\sum_{\vec{x} \in \{0,1\}^{\ell}} f(\vec{x}) = H$, for H a purported field value. The Sum-Check protocol is an interactive protocol which proves the above equality. Unfortunately, the protocol is inherently **not** zero-knowledge, as it discloses evaluations of f and f is constructed from an instance-witness pair.

To achieve a zero-knowledge Sum-Check, we must ensure that the execution transcript does not leak information about the witness or intermediate circuit wires. The masking process is divided into two distinct logical steps: hiding the circuit transcript on the hypercube, and hiding the intermediate round evaluations. We recall only the required material here and refer the reader to [Tha22, §13.3] and [XZZ⁺19] for all the details.

We begin by recalling how to mask the intermediate rounds, following the masked Sum-Check protocol devised in [XZZ⁺19, Construction 1, §4.1]. The protocol picks a sum of small random univariate polynomials $g(X_1, \dots, X_{\ell}) = a_0 + \sum_{i=1}^{\ell} g_i(X_i)$ with $g_i(X_i) = \sum_{j=1}^{d_i} a_{i,j} X_i^j$; note that here we only require that the degrees of the g_i ’s correspond to the partial degrees of f .

Protocol 5 zkSumCheck: Zero-Knowledge Sum-Check Protocol

Input: Polynomial $f(x_1, \dots, x_{\ell})$, claimed sum $H = \sum_{x \in \{0,1\}^{\ell}} f(x)$, and individual degrees $(d_1, \dots, d_{\ell})$.
Output: *Accept* or *Reject* the validity of H .

-
- (acc/rej) $\leftarrow \text{zkSumCheck}(f, H, \ell, (d_1, \dots, d_\ell))$.
-
- Step 1: Prover Commitments*
-
- 1: $\mathcal{P}$ selects a random masking polynomial $g(x_1, \dots, x_\ell) = a_0 + \sum_{i=1}^\ell g_i(x_i)$.
For each $i \in [\ell]$, $g_i(x_i) = \sum_{j=1}^{d_i} a_{i,j} x_i^j$ where $a_{i,j}$ are uniformly random.
 - 2: $\mathcal{P}$ computes $G = \sum_{x \in \{0,1\}^\ell} g(x)$ and $\text{com}_g = \text{Commit}(g)$.
 - 3: $\mathcal{P} \rightarrow \mathcal{V}$: Sends G and com_g .
-
- Step 2: Verifier Challenge*
-
- 4: $\mathcal{V}$ uniformly selects challenge $\rho \in \mathbb{F}^*$.
 - 5: $\mathcal{V} \rightarrow \mathcal{P}$: Sends ρ .
 - 6: Both parties compute the combined claim: $H' = H + \rho \cdot G$.
-
- Step 3: Interactive Sum-Check*
-
- 7: $\mathcal{P}$ and $\mathcal{V}$ execute standard Sum-Check protocol on: $H + \rho \cdot G = \sum_{x \in \{0,1\}^\ell} (f(x) + \rho g(x))$.
 - 8: At the final round, $\mathcal{V}$ obtains claim $h_\ell(r_\ell)$ for random point $r = (r_1, \dots, r_\ell)$.
-
- Step 4: Verification*
-
- 9: $\mathcal{P} \leftrightarrow \mathcal{V}$: At the final Sum-Check round, prover and verifier open the commitment of g at r : $(g(r), \pi) \leftarrow \text{Open}(g, r)$.
 - 10: $\mathcal{V}$ checks commitment: compute the $\text{Verify}(\text{com}_g, g(r), r, \pi)$. **If reject, abort.**
 - 11: $\mathcal{V}$ computes the residue: $v_f = h_\ell(r_\ell) - \rho g(r)$.
 - 12: **return** Compare v_f with oracle access of $f(r_1, \dots, r_\ell)$.
-

Since g is a uniformly random polynomial with the same degree profile as f , and the verifier challenge ρ is sampled after the prover commits to g , the polynomial $f + \rho g$ itself is *not* indistinguishable from a uniformly random polynomial of the same degree profile; note that it still agrees with f on all but the single masking term. What is masked, instead, are the Sum-Check round messages. Each per-round univariate polynomial sent by the prover is statistically close to uniform (over the randomness of g), so the transcript leaks (up to the caveat in Remark 3.18 below) no information about f or the witness beyond the final evaluation. The simulator can generate an indistinguishable transcript of round messages by sampling g and ρ as above and running the standard Sum-Check machinery on $f + \rho g$, without access to the real witness f .

Remark 3.18.

- (1) *As it stands, the Libra masking protocol is not completely zero-knowledge (cf. [Tha22, 13.3] or [CFW26]) as it reveals the evaluation of f at $\vec{r}$ (from the opening of g at $\vec{r}$). If we aim for total zero-knowledge, we must hide this value.*
- (2) *As long as the commitment scheme has opening and verification which is zero-knowledge, then the protocol is zero-knowledge (up to the final evaluation).*
- (3) *We will make use of the simulator constructed in [XZZ⁺19, Theorem 3] as a reference point, but not as our simulator. In our instantiation the mask is never opened, and the masking turns out to be perfect rather than statistical. The transcript, the final value h included, is uniform on its consistency class independently of the polynomial being summed (§4.1.1.1). We therefore simulate by sampling that class directly, at zero advantage.*

3.5. Replacing the PCS with the ABDLOP commitment scheme. For instantiating the Libra-style masking, we can either consider a multivariate polynomial commitment scheme over extension fields $\mathbb{K}$, or, more conveniently, make use of the ABDLOP proof of knowledge (cf. Protocol $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$). The natural candidates for a PCS approach would be a modification of the multilinear Hachi [NOZ26], whose generic reduction sends evaluation claims over $\mathbb{F}_{q^e}$ to claims over $\mathbb{R}_{\mathbb{F}}$, or a split PCS approach built on the univariate lattice-based scheme Greyhound [NS24]. The split-PCS route might seem attractive, but Greyhound's whole value proposition is compressing a *large* polynomial, whereas the masking polynomial p is small compared to the polynomials that genuinely need compression with a PCS: succinctness is irrelevant when the object being committed to is already logarithmic in the size of what we are proving about, and here we would pay $\approx d^2 \log m \log q$ to compress something smaller than a single Sum-Check transcript. The same objection applies to Hachi: it is a succinct scheme for large multilinear polynomials, and our p has a few coefficients. We therefore adopt the ABDLOP-based route throughout, and record both alternatives

only as points of comparison; it may still be of independent interest for larger masking polynomials in other settings. For completeness, such a split-PCS scheme can be built by combining Greyhound, which satisfies an honest-verifier zero-knowledge evaluation property, with a ring-switching embedding in the style of [FMN23, §5.6], realizing $\mathbb{K}$ as (a component of) a cyclotomic ring over which Greyhound is defined; the trick is usually performed with primes $q \equiv 5 \pmod{8}$, and extends, following [BCS23, Lemma 7.3] and the cyclotomic factorization results recalled in [Was97], to primes outside that residue class.

Concretely, recall from §3.4 that we only need to

- (1) bind p before ζ is sampled,
- (2) hide p throughout, or the Sum-Check masking fails, and
- (3) convince $\mathcal{V}$ that the value $p(\vec{r}')$ entering the final check is the evaluation of the committed p .

Nothing here requires the commitment or proof to be sublinear in p . And p is tiny: $p(\vec{X}) = a_0 + \sum_{i=1}^{\ell} \tilde{p}_i(X_i)$ has $1 + D_{\max} \cdot \ell$ coefficients in $\mathbb{K}$, which, at $b = 2$, $u \leq 2$, $m = 2^{21}$, is $1 + 4 \times 21 = 85$ elements (two ring elements of $R_{\mathbb{K}}$ at $d = 64$, and one at $d = 128$; cf. the packing count $\lceil (1 + D_{\max} \log_2(m))/d \rceil$ used in Protocol 6).

The ABDLOP commitment scheme can be used for the evaluation as well. More precisely, we pack the coefficients of p into $\lceil (1 + D_{\max} \log_2(m))/d \rceil$ ring elements and commit with $\text{Commit}_{\text{ABDLOP}}$, which takes unbounded-norm messages. Then

$$(3.13) \quad p(\vec{r}') = \langle \text{coeffs}, (1, r'_i, r_i'^2, r_i'^3, r_i'^4)_{i \in [\ell]} \rangle$$

is a *public linear function of the committed message*. This is exactly the relation $\mathbf{u} = R_1 \mathbf{s}_1 + R_m \mathbf{m}$ that Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ proves, with $R_1 = 0$ and $R_m := \sum_{i,j} (r'_i)^j \tau_{\ell(i,j)}$ the packaged rotation (Lemma 3.12, cf. §3.2.1.1) sending the coefficients of p packed into coeffs (cf. also Protocol 6) to their weighted sum $p(\vec{r}')$ in the constant slot. Note that once we have committed to p in such fashion, then we implicitly commit to the value P ; $P = \sum_{\vec{x} \in \{0,1\}^{\log_2 m}} p(\vec{x})$ is itself a public linear function of the same committed coefficient vector, recoverable from the same commitment with $R_1 = 0$ and the appropriate R_m .

Honest-verifier zero-knowledge comes from $\mathcal{S}_{\text{ABDLOP}}$ (cf. Lemma 3.6), so Remark 4.1(2)-type requirements are satisfied by the same simulator. We will see that, in practice, we do not reveal the evaluation $p(r)$, but use it inside a more general ABDLOP ZKPoK.

3.6. A brief recall on rejection sampling. Besides the changes regarding the commitments, we must modify our protocols by including a rejection sampling algorithm, in the spirit of [LNP22, Lemma 2.14]. This is of utmost importance in order to achieve a blinding folding scheme.

With the notation introduced in §§2.1–2.3, let us fix the rejection sampling parameter:

- $\mathfrak{s} := \xi \cdot (K + k) \cdot T \cdot (b - 1) \cdot \sqrt{n_{\mathbb{F}}}$ be a fixed standard deviation, and
- $\xi > 0$.

We denote by $D_{\mathfrak{s}}^{\ell}$ the discrete Gaussian distribution on $R_{\mathbb{F}}^{\ell}$ centered around 0 with standard deviation $\mathfrak{s}$ (see [LNP22, Definition 2.1]). Since we are working with infinity norms, we replace [LNP22, Lemma 2.2] (cf. also [Ban93, Lemma 1.5(i)]) by the following statement.

Lemma 3.19 ([Ban93, Mic16]). *Let $\mathbf{z} \leftarrow D_{\mathfrak{s}}^{\ell}$ for $(\mathbf{z} \in R_{\mathbb{F}}^{\ell})$. Then $\Pr[\|\mathbf{z}\| > \tau_{\lambda, \xi} \cdot \mathfrak{s}] < 2 \cdot \ell d \cdot e^{-\pi \cdot (\tau_{\lambda, \xi})^2}$.*

Note that in our setting $n_{\mathbb{F}} \approx \ell \cdot d$. Lemma 3.19 gives $\Pr[\|\mathbf{z}\| > \tau_{\lambda, \xi} \mathfrak{s}] < 2n_{\mathbb{F}} e^{-\pi \tau_{\lambda, \xi}^2}$, so a tail probability of at most $2^{-\lambda}$ requires

$$\tau_{\lambda, \xi} \geq \sqrt{(\lambda + \log_2(2n_{\mathbb{F}})) \ln 2 / \pi}.$$

At our working level $\lambda = 120$ this gives $\tau_{\lambda, \xi} \approx 5.50$ for $n_{\mathbb{F}} = 2^{16}$ and $\tau_{\lambda, \xi} \approx 5.58$ for $n_{\mathbb{F}} = 2^{20}$; at $\lambda = 128$ the corresponding values are 5.657 and 5.734. We note that k depends on λ only through this square root. The change from $\lambda = 128$ to $\lambda = 120$ alters $\tau_{\lambda, \xi}$ by under 3%.

Lemma 3.20 ([LNP22, Lemma 2.4, Figure 1]). *Let $V \subseteq R_{\mathbb{F}}^{\ell}$ be a set of polynomials with norm at most $L \cdot \sqrt{n_{\mathbb{F}}}$ and $\rho: V \rightarrow [0, 1]$ be a probability distribution. Fix the standard deviation $\mathfrak{s} = \xi \cdot L \cdot \sqrt{n_{\mathbb{F}}}$. Let $M = \exp(14/\xi + 1/(2\xi^2))$ (the constant 14 is the one derived in [LNP22, Lemma 2.14] at $\lambda = 128$; we use it unchanged at $\lambda = 120$, which is conservative). Now, sample $\mathbf{v} \leftarrow \rho$ and $\mathbf{y} \leftarrow D_{\mathfrak{s}}^{\ell}$, set $\mathbf{z} = \mathbf{y} + \mathbf{v}$. There exists a rejection sampling function Rej_1 , with $\text{Rej}_1(\mathbf{z}, \mathbf{v}, \mathfrak{s}) = 1$ signaling a restart (following the standard convention of [LNP22], so that $\text{Rej}_1 = 0$ is the accepting outcome) such that the probability that $\text{Rej}_1(\mathbf{z}, \mathbf{v}, \mathfrak{s})$ accepts lies*

in $[(1 - 2^{-\lambda})/M, 1/M]$ for every $\mathbf{v} \in V$ and the distribution of $(\mathbf{v}, \mathbf{z})$, conditioned on the acceptance of $\text{Rej}_1(\mathbf{z}, \mathbf{v}, \mathbf{s})$, is within statistical distance $2^{-\lambda}$ of the product distribution $\rho \times D_{\mathbf{s}}^\ell$.

Note that the upper bound $1/M$ is immediate from the acceptance rule $\min\{1, D_{\mathbf{s}}^\ell(\mathbf{z})/(M D_{\mathbf{s}, \mathbf{v}}^\ell(\mathbf{z}))\}$, averaging over $\mathbf{z} \leftarrow D_{\mathbf{s}, \mathbf{v}}^\ell$. Moreover, [LNP22] considers a security parameter of $\lambda = 128$ and thus, for $\xi \approx 13$, we have that the rejection repetition ends in ≈ 3 repetitions and for $\xi \approx 20$ in ≈ 2 repetitions.

In order to ensure the blinding property, we must combine the above parameters so as not to exceed the norm bound $B = b^k$, that is, $\tau_{\lambda, \xi} \cdot \mathbf{s} < B$ (equivalently $\tau_{\lambda, \xi} \xi L \sqrt{n_{\mathbb{F}}} < b^k$ with $L := (K + k)T(b - 1)$ as in Lemma 3.20). With the above notation, we must have, for $K = 1 = b - 1$ (which is our use case, cf. §4.2-),

- $k = 26$ for $T = 128$, $d = 64$, $n_{\mathbb{F}} = 2^{16}$, and $\xi \approx 13$,
- $k = 27$ for $T = 128$, $d = 64$, $n_{\mathbb{F}} = 2^{16}$, and $\xi \approx 20$,
- $k = 29$ for $T = 128$, $d = 64$, $n_{\mathbb{F}} = 2^{20}$, and $\xi \approx 13$,
- $k = 29$ for $T = 128$, $d = 64$, $n_{\mathbb{F}} = 2^{20}$, and $\xi \approx 20$,
- $k = 27$ for $T = 256$, $d = 128$, $n_{\mathbb{F}} = 2^{16}$, and $\xi \approx 13$ (this case is sensitive to λ : the bound moves to $k = 28$ for $\lambda \geq 127$, and the margin at $\lambda = 120$ is under 0.05 bits),
- $k = 28$ for $T = 256$, $d = 128$, $n_{\mathbb{F}} = 2^{16}$, and $\xi \approx 20$,
- $k = 30$ for $T = 256$, $d = 128$, $n_{\mathbb{F}} = 2^{20}$, and $\xi \approx 13$,
- $k = 30$ for $T = 256$, $d = 128$, $n_{\mathbb{F}} = 2^{20}$, and $\xi \approx 20$.

Considering Definition 3.1, we also compute:

- $k = 29$ for $T = 128$, $d = 64$, $n_{\mathbb{F}} = 2^{21}$, and $\xi \approx 13$,
- $k = 30$ for $T = 128$, $d = 64$, $n_{\mathbb{F}} = 2^{21}$, and $\xi \approx 20$,
- $k = 30$ for $T = 256$, $d = 128$, $n_{\mathbb{F}} = 2^{21}$, and $\xi \approx 13$,
- $k = 31$ for $T = 256$, $d = 128$, $n_{\mathbb{F}} = 2^{21}$, and $\xi \approx 20$.

3.7. The relations. In order to obtain zero-knowledge we must consider a few new type of relations, namely, the Norm-bounded committed CCS Evaluation Relation, the Relaxed committed CCS evaluation relation, the restricted output relation, the norm-bounded uncommitted CCS relation, and the unpadded norm-bounded CCS. We highlight in pink the differences between these relation and the one recalled in §2.4.1.

Recall that we denote by $\tilde{B} \leq B$ (actually $\tilde{B} = b$) be the fixed bound norm for the randomness of the ABDLOP commitment scheme Commit.

Definition 3.21 (Norm-bounded committed CCS Evaluation Relation). *Let $\mathbf{s}$ be a structure as defined in §2.4.1. Let $\mathcal{L} : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow \mathbb{C}$ be an arbitrary $R_{\mathbb{F}}$ -module homomorphism. Define $\mathcal{L}_{\text{in}} : R_{\mathbb{F}}^{n_{\mathbb{R}}} \rightarrow R_{\mathbb{F}}^{n_{\mathbb{R}}, \text{in}}$ to be the trivial $R_{\mathbb{F}}$ -module homomorphism that projects the first $n_{\mathbb{R}, \text{in}}$ indices. Let Commit be a non-succinct ring commitment scheme, like Commit_{BDLOP} or Commit_{ABDLOP} (recall §2.5.2) over the extension ring $R_{\mathbb{K}}$. We define the **norm-bounded CCS evaluation relation**, $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$, for $e = 1$ if Commit = Commit_{BDLOP} and $e = 2$ if Commit = Commit_{ABDLOP}, as follows:*

$$\left\{ \left(\mathbf{s}; \begin{pmatrix} c \in \mathbb{C}, \\ x \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}, \\ r \in \mathbb{K}^{\log_2(m)}, \\ \{C_j \in R_{\mathbb{F}}^{K_{\text{com}}} \}_{j \in [t]} \end{pmatrix} ; \begin{pmatrix} z \in \mathbb{F}^{n_{\mathbb{F}}}, \\ \{y_j \in R_{\mathbb{K}}\}_{j \in [t]}, \\ \{\mathbf{s}_j = (\mathbf{s}_{1,j}, \mathbf{s}_{2,j}) \in R_{\mathbb{F}}^{m_1} \times R_{\mathbb{F}}^{m_2}\}_{j \in [t]}, \end{pmatrix} \right) : \begin{array}{l} c = \mathcal{L}(\mathbf{z}) \wedge \mathbf{x} = \mathcal{L}_{\text{in}}(\mathbf{z}) \wedge \|\mathbf{z}\|_{\infty} < b \\ \wedge \|\mathbf{s}_j\|_{\infty} < \tilde{B} \\ \forall j \in [t], y_j = \widetilde{M_j \mathbf{z}}(r) \\ \wedge C_j = \text{Commit}(y_j, \mathbf{s}_j) \end{array} \right\}.$$

For the input side of Π'_{DEC} we also need the relaxed variant that a lattice proof of knowledge can certify. Recall (Lemma 3.6) that extraction from Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ returns an opening together with a challenge difference $\bar{c} \in \bar{\mathcal{C}}$, and that of its three conclusions only the norm bound carries $\bar{c}$: items (1) and (3) hold for $\bar{\mathbf{s}}$ itself. The relaxation is therefore multiplicative by a ring element rather than a looser constant, and cannot be absorbed by enlarging the bound. $\bar{c}^{-1}$ exists (Remark 2.15) but is not necessarily short.

Definition 3.22 (Relaxed committed CCS evaluation relation). *For $\beta > 0$ let $\text{CE}_{\text{com}}^{\text{rlx}}(b, \mathcal{L}, \beta, \text{Commit})$ be defined exactly as $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ in Definition 3.21, with the salt clause $\|\mathbf{s}_j\|_{\infty} < \tilde{B}$ replaced by*

$$\exists \bar{c} \in \bar{\mathcal{C}} \setminus \{0\} \text{ such that } \|\bar{c} \mathbf{s}_j\|_2 \leq \beta \quad \text{for all } j \in [t],$$

every other clause, in particular $\|\mathbf{z}\|_\infty < b$, being unchanged. We write

$$(3.14) \quad \beta_{\text{rlx}} := 2\mathfrak{s}_2 \sqrt{2N_c m_2 d}$$

for the bound Lemma 3.6(2) supplies at the concatenation length N_c of the call in question, $\mathfrak{s}_2$ denoting the width actually used by that call.

Note that $\mathfrak{s}_2 = \mathfrak{s}_2^{\text{cat}} = \gamma_2 \eta \nu \sqrt{N_c m_2 d}$ of Definition 3.15 when the call is a concatenated one, and the single-block width otherwise. The distinction is worth stating, since N_c then enters β_{rlx} twice, once through the width and once through the dimension $N_c m_2 d$ of the randomness vector: reading $\mathfrak{s}_2$ as the single-block width would understate β_{rlx} by a factor $\sqrt{N_c}$, and β_{rlx} is the bound at which binding is assumed.

Remark 3.23 ($\mathcal{R}_2 \subseteq \mathcal{R}'_2$ at the $\Pi'_{\text{RLC}}/\Pi'_{\text{DEC}}$ interface). *Fix any $\bar{c} \in \bar{\mathcal{C}} \setminus \{0\}$. If $\|\mathbf{s}_j\|_\infty < B_{\text{fold}}$ then $\|\bar{c}\mathbf{s}_j\|_\infty \leq 2T(\mathcal{C})B_{\text{fold}}$ and hence $\|\bar{c}\mathbf{s}_j\|_2 \leq 2T(\mathcal{C})B_{\text{fold}}\sqrt{(m_1 + m_2)d}$, so*

$\text{CE}_{\text{com}}(B, \mathcal{L}, B_{\text{fold}}, \text{Commit}) \subseteq \text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit})$ whenever $\beta_{\text{rlx}} \geq 2T(\mathcal{C})B_{\text{fold}}\sqrt{(m_1 + m_2)d}$, which holds comfortably at the width $\mathfrak{s}_2$ of Step 5 of Protocol 8. Completeness of Π'_{RLC} is therefore unaffected by reading its output in the relaxed relation, and we use the two interchangeably in the sense of the notation $\mathcal{R}_2(\mathcal{R}'_2)$ of Proposition 2.8. The tight relation records what an honest execution produces, the relaxed one what extraction certifies.

Note that binding is not weakened. Since every nonzero element of $\bar{\mathcal{C}} = \mathcal{C} - \mathcal{C}$ is invertible in $\mathbb{R}_{\mathbb{F}}$ (Remark 2.15), two accepting relaxed openings of one commitment to distinct messages still yield $A_1 \Delta \mathbf{s}_1 + A_2 \Delta \mathbf{s}_2 = 0$ with $0 \neq \|(\Delta \mathbf{s}_1, \Delta \mathbf{s}_2)\|_2 \leq 2\beta_{\text{rlx}}$, i.e. a solution to $\text{MSIS}_{\kappa, m_1 + m_2, 2\beta_{\text{rlx}}}$. The only effect is on the MSIS bound at which binding is assumed, which is already the bound governing the proofs of knowledge invoked in Protocols 6 and 7.

For the interface between Π'_{R1CS} and Π'_{RLC} we need one further relaxation, because the extractor of Lemma 4.10 inverts a challenge difference coordinate-wise (Lemma 2.19) on top of the $\bar{c}$ that Lemma 3.6(2) already supplies. Write

$$\bar{\mathcal{C}}^{\cdot e} := \{\bar{c}_1 \cdots \bar{c}_e : \bar{c}_1, \dots, \bar{c}_e \in \bar{\mathcal{C}} \setminus \{0\}\}, \quad T_1(\mathcal{C}) := \max_{\bar{c} \in \bar{\mathcal{C}} \setminus \{0\}} \|\bar{c}\|_1,$$

so that $\|\bar{c}\mathbf{v}\|_2 \leq T_1(\mathcal{C})\|\mathbf{v}\|_2$ for every $\bar{c} \in \bar{\mathcal{C}}$ and every $\mathbf{v}$, the ring being negacyclic. Every element of $\bar{\mathcal{C}}^{\cdot e}$ is a product of invertibles (Remark 2.15) and is in particular nonzero.

Definition 3.24 (e -fold relaxed committed CCS evaluation relation). *For $\beta > 0$ and $e \in \mathbb{Z}_{\geq 1}$ let $\text{CE}_{\text{com}}^{\text{rlx}, e}(b, \mathcal{L}, \beta, \text{Commit})$ be defined exactly as $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ in Definition 3.21, with the salt clause $\|\mathbf{s}_j\|_\infty < \tilde{B}$ replaced by*

$$\exists \Delta \in \bar{\mathcal{C}}^{\cdot e} \text{ such that } \|\Delta \mathbf{s}_j\|_2 \leq \beta \quad \text{for all } j \in [t],$$

every other clause unchanged. Thus $\text{CE}_{\text{com}}^{\text{rlx}, 1} = \text{CE}_{\text{com}}^{\text{rlx}}$ of Definition 3.22. We write

$$(3.15) \quad \beta_{\text{rlx}}^{(3)} := 2T_1(\mathcal{C})\beta_{\text{rlx}}.$$

Two inclusions are used below and both are immediate. First, $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}) \subseteq \text{CE}_{\text{com}}^{\text{rlx}, 3}(q/2, \mathcal{L}, \beta_{\text{rlx}}^{(3)}, \text{Commit})$: fix any $\bar{c} \in \bar{\mathcal{C}} \setminus \{0\}$ and take $\Delta := \bar{c}^3$, so that $\|\Delta \mathbf{s}_j\|_2 \leq T_1(\mathcal{C})^2 2T(\mathcal{C})\tilde{B}\sqrt{(m_1 + m_2)d}$. Second, $\text{CE}_{\text{com}}^{\text{rlx}, 3}(q/2, \mathcal{L}, \beta_{\text{rlx}}^{(3)}, \text{Commit}) \subseteq \text{CE}_{\text{com}}(q/2, \mathcal{L}, q/2, \text{Commit})$ strictly, the latter imposing no condition on $\mathbf{s}_j$ at all beyond the commitment identity. The point of the definition is exactly this: it is the smallest relation that both contains the honest output and records what extraction certifies.

We can notice that we have a canonical (surjective) projection

$$(3.16) \quad \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}) \longrightarrow \text{CE}(b, \mathcal{L})$$

defined by forgetting the extra commitment structure and making the y_j values public:

$$(3.17) \quad \left(\mathbf{s}; \left(\begin{array}{c} c \in \mathbb{C}, \\ x \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}, \\ r \in \mathbb{K}^{\log_2(m)}, \\ \{C_j \in \mathbb{R}_{\mathbb{F}}^{\kappa_{\text{com}}}\}_{j \in [t]} \end{array} \right); \left(\begin{array}{c} z \in \mathbb{F}^{n_{\mathbb{F}}}, \\ \{y_j \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]} \end{array} \right) \right) \mapsto \left(\mathbf{s}; \left(\begin{array}{c} c \in \mathbb{C}, \\ x \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}, \\ r \in \mathbb{K}^{\log_2(m)}, \\ \{y_j \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]} \end{array} \right); (z \in \mathbb{F}^{n_{\mathbb{F}}}) \right).$$

In order to achieve our zero-knowledge goal, we need to further restrict ourselves to a finer relation. For the sake of simplicity we sometimes present the protocols in §4.1 in terms of Definition 3.21, and only consider

the following Definition 3.25 in §4.2; this is the set in which both an honest execution (Lemma B.1 and §3.6) and the simulator in the proof of Theorem 4.13 land.

Definition 3.25 (The restricted output relation). *We write $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$ for the subset of $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ consisting of those tuples whose recomposition $\mathbf{z} := \sum_{i \in [k]} b^{i-1} \mathbf{z}_i$ satisfies*

$$\|\mathbf{z}\|_{\infty} \leq \tau_{\lambda, \xi} \cdot \mathfrak{s}.$$

In the end, we fix the following (last) relation.

Definition 3.26 (Norm-bounded uncommitted CCS). *We write $\text{CCS}(b)^K$ for*

$$\left\{ (s; (x \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}); w \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}, \text{in}}}) : \begin{array}{l} \text{For } \mathbf{z} := [x, w], \\ \|\mathbf{z}\|_{\infty} < b \wedge f(\widetilde{M_1 \mathbf{z}}, \dots, \widetilde{M_t \mathbf{z}}) \in \text{ZS}_{\log_2(m)} \end{array} \right\}$$

We will consider the embedding (cf. also Definition 3.1)

$$(3.18) \quad \text{CCS}(b)^K \subset \text{CCS}(b, \mathcal{L})^K$$

defined by applying the commitment $\mathcal{L}$.

Unfortunately, this is still not enough for suitably defining a NovaBlindFold analogue.

Remark 3.27. Recall Definition 2.11. It hands the instances (u_1, u_2) to $\mathcal{S}$ as input, so it asserts nothing about them, and it says that the interaction leaks nothing beyond the public instances. Whether that is enough for zero-knowledge therefore depends entirely on which relation one takes $\mathcal{R}'$ to be.

Read against $\mathcal{R}' = \text{CCS}(b, \mathcal{L})$, it is not enough, and for two reasons at once. First, $u_2 = (c_1, \mathbf{x}_1)$ with $c_1 = \mathcal{L}(\mathbf{z}_1)$, and $\mathcal{L}$ is a compact Ajtai commitment carrying no salt, so c_1 is a deterministic function of the witness; handing it to $\mathcal{S}$ is handing over the very object the blinded layout exists to hide. Second, $\mathcal{A}$ would output w_2 in full, blinding block included, so an $\mathcal{A}$ setting $\mathbf{e} := 0$ fixes c_1 outright and Lemma 3.3 has no hypothesis left to work with.

Both defects disappear when we cut the input relation correctly. We will see that neither c_1 nor $\mathbf{e}$ is in fact an input to Π_{LBF} : the protocol takes its input uncommitted (Definition 3.26) and forms c_1 itself, and Definition 3.1.(1) has the honest prover sample $\mathbf{e}$. Treating either as part of u_2 describes a protocol other than the one we run. Taking $\mathcal{R}'$ to be the unpadding relation of Definition 3.28 (see below) puts both objects where they belong, that is, $\mathbf{e}$ in the prover's randomness and c_1 in the transcript. Definition 2.11 may then be applied verbatim.

Definition 3.28 (Unpadding norm-bounded CCS). *Let $\mathbf{s}$ be an RICS structure in the blinded layout of Definition 3.1, with blinding coordinates $\mathcal{I}_{\text{bl}}$ of ring-length $n_{\text{R}, \text{bl}}$, and write $\mathbf{s}^{\circ}$ for the structure obtained by deleting the rows and columns indexed by $\mathcal{I}_{\text{bl}}$, so that $\mathbf{s} = \text{pad}(\mathbf{s}^{\circ})$ for the deterministic, public padding map of that definition. We write $\text{CCS}^{\circ}(b)^K$ for the relation $\text{CCS}(b)^K$ of Definition 3.26 taken over $\mathbf{s}^{\circ}$: its instances are the public inputs x , and its witnesses the circuit part $\mathbf{w}_{\text{circ}}$ of the witness alone, with no blinding block.*

The two relations are related by a randomised prover-side map

$$(3.19) \quad \iota_{\text{bl}}: \text{CCS}^{\circ}(b)^K \longrightarrow \text{CCS}(b)^K \subset \text{CCS}(b, \mathcal{L})^K, \quad (\mathbf{s}^{\circ}; x; \mathbf{w}_{\text{circ}}) \longmapsto (\text{pad}(\mathbf{s}^{\circ}); x; [\mathbf{w}_{\text{circ}}, \mathbf{e}]),$$

with $\mathbf{e} \leftarrow \chi_b^{n_{\text{R}, \text{bl}}}$ the block sampled by the honest prover, χ_b uniform on $\{c \in \mathbb{R}_{\mathbb{F}} : \|c\|_{\infty} \leq b - 1\}$ as in Lemma 3.3, and the second inclusion the deterministic one of Equation (3.18), given by applying $\mathcal{L}$. Everything we need of ι_{bl} follows from one structural observation.

Lemma 3.29. *Let $\mathbf{s} = \text{pad}(\mathbf{s}^{\circ})$ follow the blinded layout of Definition 3.1, and fix a public input x . Write $\mathbf{z}^{\circ} := [x, \mathbf{w}_{\text{circ}}]$ and $\mathbf{z} := [x, \mathbf{w}_{\text{circ}}, \mathbf{e}]$. Then*

$$(s; x; [\mathbf{w}_{\text{circ}}, \mathbf{e}]) \in \text{CCS}(b) \iff (\mathbf{s}^{\circ}; x; \mathbf{w}_{\text{circ}}) \in \text{CCS}^{\circ}(b) \text{ and } \|\mathbf{e}\|_{\infty} < b.$$

Equivalently, the fibre of $\text{CCS}(b)$ over a given $(x; \mathbf{w}_{\text{circ}})$ is exactly $\{\mathbf{e} : \|\mathbf{e}\|_{\infty} < b\}$, the same set for every $(x; \mathbf{w}_{\text{circ}})$.

Proof. The norm condition splits coordinatewise, $\|\mathbf{z}\|_{\infty} < b$ iff $\|\mathbf{z}^{\circ}\|_{\infty} < b$ and $\|\mathbf{e}\|_{\infty} < b$. For the predicate, $f(\widetilde{M_1 \mathbf{z}}, \dots, \widetilde{M_t \mathbf{z}}) \in \text{ZS}_{\log_2(m)}$ asks that the row values $f((M_1 \mathbf{z})_i, \dots, (M_t \mathbf{z})_i)$ vanish for every i , so we may treat the rows separately. For $i \in \mathcal{I}_{\text{bl}}$, clause (3) of Definition 3.1 makes row i read $z_i \cdot 1 - z_i = 0$, which

holds whatever $\mathbf{e}$ is (Remark 3.2). For $i \notin \mathcal{I}_{\text{bl}}$, clause (4) gives $(M_j)_{i, \mathcal{I}_{\text{bl}}} = 0$, whence $(M_j \mathbf{z})_i = (M_j^\circ \mathbf{z}^\circ)_i$ for every j , so row i evaluates identically under $\mathbf{s}$ and under $\mathbf{s}^\circ$. The circuit rows therefore hold for $\mathbf{z}$ iff they hold for $\mathbf{z}^\circ$, and the blinding rows hold always. ■

We have three consequences, in the order we use them.

- ι_{bl} is total: it is defined on the whole of $\text{CCS}^\circ(b)^K$ and succeeds on every draw, never returning $\perp$. What gives this is not merely that the fibre is non-empty but that it *coincides* with the support of the sampler. Lemma 3.29 identifies the fibre with $\{\mathbf{e} : \|\mathbf{e}\|_\infty < b\}$, which is exactly the support of $\chi_b^{n_{\text{R}, \text{bl}}}$, so every value the prover can draw is a valid completion.
- *Truncation is well defined*: Deleting the block is a map $\pi : \text{CCS}(b) \rightarrow \text{CCS}^\circ(b)$ with $\pi \circ \iota_{\text{bl}} = \text{id}$, so composing any extractor for Protocol 12) with π extracts for the final composition Π_{LBF} , at the same advantage and with no rewinding beyond what the former already does. Completeness and knowledge soundness transfer along ι_{bl} unchanged; cf. Propositions 4.15 and 4.17.
- *The fibre does not depend on the circuit witness*: $\mathbf{e}$ is drawn from a set fixed in advance by b and $n_{\text{R}, \text{bl}}$ alone, so its distribution is independent of anything an adversary $\mathcal{A}$ may choose.

4. LATTICEBLINDFOLD

We are now ready to assemble our LatticeBlindFold protocol Π_{LBF} . We rework each of SuperNeo's three interactive reductions (Π_{CCS} , Π_{RLC} , and Π_{DEC}) into its zero-knowledge counterpart, applying the tools recalled and devised in §3.

4.1. The building blocks of LatticeBlindFold. We highlight in **purple** the changes we introduce regarding the masking technique, in **pink** the change of relation, in **blue** the modifications related to the commitments, and in **orange** the ones related to the choice of independent challenges.

Remark 4.1.

- (1) We focus on the case of R1CS structures, that is, f has degree $\deg f = u \leq 2$; however the whole procedure can be generalized to handle CCS structures. The drawback in handling general structures is that the amount of commitments grows accordingly, quickly rendering the scheme impractical.
- (2) For brevity we denote by **Commit** the commitment $\text{Commit}_{\text{ABDLOP}}$ in all the following sections, and by $\tilde{B}$ the coefficient bound from which commitment salts are sampled. We take $\tilde{B} = 2$ (ternary salts) throughout, purely for concreteness. The relevant salt bounds are B_{fold} of Lemma B.1, on the folded salt, and β_{rlx} of Definition 3.22, on extracted openings, both independent of k .
- (3) We assume the number of constraints to be equal to the size of the instance-witness $m = n_{\mathbb{F}}$ a power of two with $d \mid n_{\mathbb{F}}$, so that the witness is a ring vector of length $n_{\text{R}} = n_{\mathbb{F}}/d$ and $M_j \in \mathbb{F}^{m \times m}$. We write the R1CS predicate as $f(Y_1, Y_2, Y_3) = Y_2 \cdot Y_3 - Y_1$ and take $M_1 = I_m$, which is without loss of generality for constraint systems arising from circuits (as every multiplication gate owns its output wire). We assume throughout that $\mathbf{s}$ and the witnesses follow the blinded layout of Definition 3.1. That is, the last $n_{\mathbb{F}, \text{bl}}$ rows are pass-through gates imposing no constraint, while $M_1 = I_m$ and the coverage of NC are preserved (Remark 3.2).

4.1.1. First Reduction: R1CS Reduction.

Protocol 6 R1CS reduction $\Pi'_{\text{R1CS}} - b = 2$ small, R1CS structure; Libra-style zk-Sum-Check.

Parameters: Refer to Definition 2.2.

Input: $\in \text{CCS}(b, \mathcal{L})^K \times \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$

$(\mathbf{s}; (c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}); w_i \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}, \text{in}}})_{i=1}^K,$
 $(\mathbf{s}; c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}, r' \in \mathbb{K}^{\log_2(m)}, \{\text{Commit}(y'_{i,j}, s'_{i,j}) \in \mathbb{R}_{\mathbb{F}}^{K_{\text{com}}}\}_{j \in [t]}; \{y'_{i,j} \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, \{s'_{i,j} \in \mathbb{R}_{\mathbb{F}}^{m_1} \times \mathbb{R}_{\mathbb{F}}^{m_2}\}_{j \in [t]}, z_i \in \mathbb{F}^{n_{\mathbb{F}}})_{i \in [K+k]}$

Output: $\in \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^{K+k}$

$(\mathbf{s}; c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}, r' \in \mathbb{K}^{\log_2(m)}, \{\text{Commit}(y'_{i,j}, s'_{i,j}) \in \mathbb{R}_{\mathbb{F}}^{K_{\text{com}}}\}_{j \in [t]}; \{y'_{i,j} \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, \{s'_{i,j} \in \mathbb{R}_{\mathbb{F}}^{m_1} \times \mathbb{R}_{\mathbb{F}}^{m_2}\}_{j \in [t]}, z_i \in \mathbb{F}^{n_{\mathbb{F}}})_{i \in [K+k]}$

Setup $\mathcal{G}(1^\lambda, n_{\mathbb{R}}) \rightarrow \text{pp}$: Output $\text{pp} \leftarrow \text{Setup}(1^\lambda, n_{\mathbb{R}})$.

Encoder $\mathcal{K}(\text{pp}, \mathbf{s}) \rightarrow (\text{pk}, \text{vk})$: Output $((\text{pp}, \mathbf{s}), \perp)$.

Reduction $\langle \mathcal{P}, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, w_1) \rightarrow (u_2, w_2)$:

- 1: $\mathcal{V} \leftrightarrow \mathcal{P}$: The prover and the verifier engage in the proof of knowledge for $\{\text{Commit}(y_{i,j}, \mathbf{s}_{i,j})\}_{i \in \{K+1, \dots, K+k\}, j \in [t]}$ as in Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$; cf. also §2.5.2.
- 2: $\mathcal{V} \rightarrow \mathcal{P}$: Send challenges $\alpha \xleftarrow{\$} \mathbb{K}^{\log_2(m)}$, $\vec{\gamma}^{(1)} \xleftarrow{\$} \mathbb{K}^{K+k}$, $\vec{\gamma}^{(2)} \xleftarrow{\$} \mathbb{K}^t$, $\vec{\gamma}^{(3)} \xleftarrow{\$} \mathbb{K}^d$, $(\delta_0, \delta_1) \xleftarrow{\$} \mathbb{K}^2$ to $\mathcal{P}$.
- 3: $\mathcal{P}$: For all $i \in [K]$, define $z_i := [x_i, w_i]$.
 Define $\vec{X} := (X_1, \dots, X_{\log_2(m)})$,

$$\text{F}(\vec{X}) := \sum_{i=1}^K \gamma_i^{(1)} \cdot f(\widetilde{M_1 z_i}, \dots, \widetilde{M_t z_i}) \in \mathbb{K}[\vec{X}] \quad \triangleright \text{Recall that } f \text{ is defining the RICS structure.}$$

$$\text{NC}(\vec{X}) := \sum_{i=1}^{K+k} \gamma_i^{(1)} \cdot \prod_{j=-(b-1)}^{b-1} (\widetilde{z_i} - j) \in \mathbb{K}[\vec{X}]$$

$$\text{Eval}(\vec{X}) := \text{eq}(\vec{X}, r) \cdot \sum_{i=K+1}^{K+k} \sum_{j=1}^t \sum_{\ell=1}^d \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)} \cdot \text{cf}(\widetilde{M_j z_i})_\ell \in \mathbb{K}[\vec{X}]$$
 where $\text{cf}(\widetilde{M_j z_i})_\ell$ is the multi-linear extension of the ℓ -th coefficient vector of $\widetilde{M_j z_i}$.
 Define $Q(\vec{X}) := \text{eq}(\vec{X}, \alpha) \cdot (\text{F}(\vec{X}) + \delta_0 \cdot \text{NC}(\vec{X})) + \delta_1 \cdot \text{Eval}(\vec{X}) \in \mathbb{K}[\vec{X}]$
 Define claimed sum of Q over $\{0, 1\}^{\log_2(m)}$ as $T := \delta_1 \cdot \sum_{i=K+1}^{K+k} \sum_{j=1}^t \sum_{\ell=1}^d \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)} \cdot \text{cf}(y_{i,j})_\ell \in \mathbb{K}$.
 $\mathcal{P}$ construct the polynomial $p(X) := a_0 + \sum_{i=1}^\ell p_i(X_i)$ where $\ell = \log_2(m)$ and, for each $i \in [\ell]$, $p_i(X_i) = \sum_{j=1}^{d_i} a_{i,j} X_i^j$ where the $a_{i,j}$ are sampled uniformly random; here $d_i := \deg_{X_i}(Q(X))$.
 4: $\mathcal{P} \rightarrow \mathcal{V}$: packs the coefficients $\{a_{i,j}\}$ of p into $\lceil (1 + D_{\max} \log_2(m))/d \rceil$ ring elements $\text{coeffs} \in \mathbb{R}_{\mathbb{K}}$ and commits via $\text{Commit}_{\text{ABDLOP}}(\text{coeffs}, \mathbf{s}_p)$, sending the commitments to $\mathcal{V}$. $\triangleright P = \sum_x p(x)$ is a public linear function of these commitments; cf. §3.5.
 5: $\mathcal{V} \rightarrow \mathcal{P}$: $\mathcal{V}$ samples a random $\zeta \xleftarrow{\$} \mathbb{K}^\times$ and sends it to $\mathcal{P}$.
 6: $\mathcal{P} \rightarrow \mathcal{V}$: The prover sends the claimed Sum-Check sum $\sigma := \zeta \cdot P + T \in \mathbb{K}$. $\triangleright \sigma$ is uniform in $\mathbb{K}$ (Remark 4.2.(7)), so T is perfectly hidden.
 7: $\mathcal{V} \leftrightarrow \mathcal{P}$: The prover and the verifier engage in the anchoring proof of knowledge Π^{anc} (§3.2.1.1) on the $N_c = 1 + kt$ commitments $\text{Commit}(\text{coeffs}, \mathbf{s}_p)$ and $\{\text{Commit}(y_{i,j}, \mathbf{s}_{i,j})\}_{i \in [K+1, K+k], j \in [t]}$, certifying the single relation

$$\text{ct}\left(\zeta R_m^{(\Sigma)} \text{coeffs} + \delta_1 \sum_{i=K+1}^{K+k} \sum_{j=1}^t \pi_{i,j} y_{i,j} - \sigma\right) = 0,$$

where $R_m^{(\Sigma)}$ is the packaged rotation (Lemma 3.12) realising $P = \sum_{\vec{x} \in \{0,1\}^{\log_2(m)}} p(\vec{x})$ as a public linear functional of coeffs , and $\pi_{i,j} := \sum_{\ell=1}^d \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)} \tau_\ell$. $\triangleright$ This is one instance of Protocol 4 with $N = 1$, $\mathbf{R}_1 = 0$ and public target $\mathbf{u} := \sigma$. No commitment is recomputed. This step is what binds T to the initial $y_{i,j}$.

- 8: $\mathcal{V} \leftrightarrow \mathcal{P}$: Prover and verifier engage in the $\text{SumCheck}(\zeta \cdot P + T; \zeta \cdot p + Q, r')$, for the verifier's sampled $r' \xleftarrow{\$} \mathbb{K}^{\log_2(m)}$, which reduces the claim that

$$\zeta \cdot P + T = \sum_{\vec{x} \in \{0,1\}^{\log_2(m)}} (\zeta \cdot p + Q)(\vec{x})$$

to the new evaluation claim $v \stackrel{?}{=} (\zeta \cdot p + Q)(r')$ for new evaluation point $r' = (r'_i)_{i=1}^{\log_2(m)} \in \mathbb{K}^{\log_2(m)}$.

In the last step of the Sum-Check interaction, $\mathcal{V}$ obtains the claim $h := h_{\log_2(m)}(r'_{\log_2(m)})$ (the prover's last-round Sum-Check message, itself sent in the clear as usual for Sum-Check, evaluated at the public point r'), matching the notation of Protocol 5; by construction of the masked Sum-Check, $h = (\zeta p + Q)(r')$. $\triangleright h$ is public, unlike $p(\vec{r}')$ and $Q(\vec{r}')$, which $\mathcal{V}$ never obtains; see §4.1.1.1.

- 9: $\mathcal{P}$: Commit to the values $c'_{i,j} := \text{Commit}(y'_{i,j}) \leftarrow \text{Commit}(\widetilde{M_j z_i})(r') \in \mathbb{R}_{\mathbb{F}}^{\text{com}}$ (recall that $y'_{i,j} \in \mathbb{R}_{\mathbb{K}}$) $\forall i \in [K+k], \forall j \in [t]$, and send them to the verifier.
- 10: $\mathcal{P} \rightarrow \mathcal{V}$: For each $i \in [K+k], j \in [t]$, sample a fresh salt and commit to a degree-0 ring element $\hat{y}_{i,j} \in \mathbb{R}_{\mathbb{K}}$ with $\text{cf}(\hat{y}_{i,j})_\ell = 0$ for all $\ell \geq 2$ and $\text{ct}(\hat{y}_{i,j}) := \text{ct}(y'_{i,j})$, via $\hat{c}_{i,j} := \text{Commit}(\hat{y}_{i,j})$, and send $\{\hat{c}_{i,j}\}$ to the verifier. $\triangleright \hat{y}_{i,j}$ carries the scalar $\text{ct}(y'_{i,j})$ needed by c_F, c_N : the Step 15 identities require degree-0 factors (Lemma 2.20), whereas c_E sums over all $\ell \in [d]$ and must route through the full hint. Cf. §3.2.1.1.
- 11: $\mathcal{V} \rightarrow \mathcal{P}$: Sample random $\{\mu_\ell\}_{\ell=2}^d \xleftarrow{\$} \mathbb{K}^{d-1}$, common to every (i, j) , and send them to $\mathcal{P}$; set $\pi_\mu := \sum_{\ell=2}^d \mu_\ell \tau_\ell \in \mathbb{R}_{\mathbb{K}}$, the packaged rotation of Lemma 3.12 (with $w_1 = 0$, $w_\ell = \mu_\ell$ for $\ell \geq 2$), so that

$\text{ct}(\pi_\mu \cdot a) = \sum_{\ell=2}^d \mu_\ell \text{cf}(a)_\ell$ for any $a \in \mathbb{R}_\mathbb{K}$. $\triangleright \pi_\mu$ collapses the $d-1$ conditions $\text{cf}(\hat{y}_{i,j})_\ell = 0$, $\ell \geq 2$, into one constant-term relation per $\hat{y}_{i,j}$ (Lemma 3.13).

- 12: $\mathcal{V} \leftrightarrow \mathcal{P}$: The prover and the verifier engage in a single $\Pi_{\text{many}}^{(\text{ct})}$ proof of knowledge (Protocol 4) for the family of $N = 2(K+k)t$ constant-coefficient relations, one per $(i,j) \in [K+k] \times [t]$ and per type:

$$\begin{aligned} \text{ct}(\pi_\mu \cdot \hat{y}_{i,j}) &= 0 & \forall i \in [K+k], \forall j \in [t], \\ \text{ct}(\hat{y}'_{i,j} - \hat{y}_{i,j}) &= 0 & \forall i \in [K+k], \forall j \in [t]. \end{aligned}$$

$\triangleright$ Constant-term relations, not ring identities, so $\Pi_{\text{many}}^{(1)}$ cannot be invoked directly (§3.2.1); Protocol 4 aggregates all N into one full-ring row. As at Step 15 the messages sit in different commitments, so the call is made on the concatenated instance of Definition 3.15. Soundness cost $2/|\mathbb{K}|$: §4.1.1.2.

- 13: $\mathcal{P}$: Commit to the values $\hat{\mathcal{C}}'_{i,sq} := \text{Commit}(\hat{y}_{i,1} \cdot \hat{y}_{i,1}) \forall i \in [K+k]$, and $\hat{\mathcal{C}}'_{i,cube} := \text{Commit}(\hat{y}_{i,1} \cdot \hat{y}_{i,1} \cdot \hat{y}_{i,1}) \forall i \in [K+k]$ and send them to the verifier. $\triangleright$ Auxiliary commitment to the intermediate square, letting the cube be chained as two quadratic relations.

- 14: $\mathcal{P} \rightarrow \mathcal{V}$: Commit to the values $\forall i \in [K]$, $\hat{\mathcal{C}}'_{i,prod} := \text{Commit}(\hat{y}_{i,2} \cdot \hat{y}_{i,3})$ and send them to the verifier. $\triangleright$ Uses the R1CS structure $f(Y_1, Y_2, Y_3) = Y_2 Y_3 - Y_1$ with $M_1 = \mathbf{I}_m$, so slot $j = 1$ is the witness itself.

- 15: $\mathcal{V} \leftrightarrow \mathcal{P}$: The prover and the verifier engage in the quadratic-relations proof of knowledge $\Pi_{\text{ABDLOP}, \text{many}}^{(2)}$ (Protocol $\Pi_{\text{many}}^{(1)}$ $((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$) with $k = 1$, $\sigma = \text{id}$ on $\hat{y}$, for the batch of $N = 2(K+k) + K$ quadratic relations, stated as identities in $\mathbb{R}_\mathbb{K}$ between the committed messages $m_{sq,i}, m_{cube,i}, m_{prod,i}$ underlying $\hat{\mathcal{C}}'_{i,sq}, \hat{\mathcal{C}}'_{i,cube}, \hat{\mathcal{C}}'_{i,prod}$ and the degree-0 surrogates $\hat{y}_{i,j}$ of Step 10:

$$\begin{aligned} m_{sq,i} - \hat{y}_{i,1} \cdot \hat{y}_{i,1} &= 0 & \forall i \in [K+k], \\ m_{cube,i} - m_{sq,i} \cdot \hat{y}_{i,1} &= 0 & \forall i \in [K+k], \\ m_{prod,i} - \hat{y}_{i,2} \cdot \hat{y}_{i,3} &= 0 & \forall i \in [K]. \end{aligned}$$

Since the three messages of each relation sit in *different* commitments, the call is made on the concatenated instance of Definition 3.15, at concatenation length N_c (and masking width $\mathfrak{s}_2^{\text{cat}}$). $\triangleright$ Equivalent to $\text{ct}(\hat{\mathcal{C}}')$ -identities, since the $\hat{y}_{i,j}$ are degree-0 by Step 12 and degree-0 elements are closed under products and differences (Lemma 2.20). Here $N_c = 3(K+k) + 3K$ and $\mathfrak{s}_2^{\text{cat}} = \mathfrak{s}_2 \sqrt{N_c}$, as in Equation (3.14). These identities certify the scalars c_F, c_N below.

- 16: $\mathcal{V}$: Derive, from commitments already in its view, the *salt-offset* intermediate evaluations, that is, the constant coefficients of the corresponding BDLOP components (writing $\text{Commit}(m, \mathbf{s}_1, \mathbf{s}_2)_2 = \mathbf{B}\mathbf{s}_2 + m$),

$$c_F := \sum_{i=1}^K \gamma_i^{(1)} \cdot \text{ct}(\text{Commit}(f(\hat{y}_{i,1}, \dots, \hat{y}_{i,t}))_2) \in \mathbb{K}$$

$$c_N := \sum_{i=1}^{K+k} \gamma_i^{(1)} \cdot \text{ct}\left(\text{Commit}\left(\prod_{j=-b}^{b-1} (\hat{y}_{i,1} - j)\right)_2\right) \in \mathbb{K}$$

$$c_E := \text{eq}(r', r) \sum_{i=K+1}^{K+k} \sum_{j=1}^t \sum_{\ell=1}^d \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)} \cdot \text{cf}(\text{Commit}(\hat{y}'_{i,j})_2)_\ell \in \mathbb{K}$$

$\triangleright c_F, c_N, c_E$

are not $F(\vec{r}')$, $\text{NC}(\vec{r}')$, $\text{Eval}(\vec{r}')$: each is that evaluation plus an aggregate of the salts of Steps 9–15, e.g. $c_F = F(\vec{r}') + \text{ct}(\mathbf{B}\mathbf{s}_{2,F})$. Those salts are never opened, so no evaluation is disclosed; Step 18 accounts for the offset. Cf. §4.1.1.1.

- 17: $\mathcal{V}$: Compute the target $u^* := h - \text{eq}(r', \alpha) \cdot (c_F + \delta_0 \cdot c_N) - \delta_1 \cdot c_E$. $\triangleright u^*$ is computed by $\mathcal{V}$ from its own view, not transmitted. By $h - \zeta p(\vec{r}') = Q(\vec{r}')$ and the Step 16 offsets, $u^* = \zeta p(\vec{r}') - \text{ct}(\mathbf{B}\mathbf{s}_{2,Q})$ for the unopened salt aggregate $\mathbf{s}_{2,Q}$.

- 18: $\mathcal{V} \leftrightarrow \mathcal{P}$: The prover and the verifier engage in a single anchoring proof of knowledge Π^{anc} (§3.2.1.1) for the verifier-computed target u^* , taken with weights $\{w_n\}$ as follows: $+\text{eq}(r', \alpha) \gamma_i^{(1)}$ on $\{s_{2,i,prod}\}_{i \in [K]}$ and $-\text{eq}(r', \alpha) \gamma_i^{(1)}$ on $\{s_{2,i,1}\}_{i \in [K]}$ (since $f(Y_1, Y_2, Y_3) = Y_2 Y_3 - Y_1$ and $\mathbf{M}_1 = \mathbf{I}_m$, so the subtracted slot is $j = 1$, exactly as for NC), the same pair scaled by δ_0 on $\{s_{2,i,cube}, s_{2,i,1}\}_{i \in [K+k]}$, $\delta_1 \text{eq}(r', r) \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)}$ on $\{s_{2,i,j}\}_{i \in [K+1, K+k], j \in [t]}$, and $\zeta \cdot R_m^{(r')}$ on $\mathbf{s}_{P,2}$ (the packaged rotation, cf. §3.5, sending coeffs's salts to the constant slot of $p(\vec{r}')$). By Lemma 3.10 this is the single constant-coefficient relation $\text{ct}(\sum_n w_n \mathbf{m}_{i(n)}^{(n)} - \mathbf{u}) = 0$ in the committed messages, with $\mathbf{u} = \sum_n w_n (\mathbf{t}_B^{(n)})_{i(n)} - u^*$ public. $\triangleright$ Accepting this proof is the evaluation check $h - \zeta p(\vec{r}') \stackrel{?}{=} Q(\vec{r}')$ at the level of the commitments. Unlike Step 7, this call uses the substitution, u^* being derived rather than transmitted.

Before proceeding further we discuss the following remarks.

Remark 4.2.

- (1) Π'_{R1CS} does not by itself certify the hint commitments. Soundness of the $\{\text{Commit}(y'_{i,j}, s'_{i,j})\}$ is discharged by membership of w_2 in $\mathcal{R}'_2$, not by Protocol 6. This is soundness relative to $\mathcal{R}'_2$ in the sense of [NS26], together with the two-run witness-agreement condition, and is exactly what Lemma 4.9 establishes; it should not be read as a failure of soundness.
- (2) As noticed above, we use a commitment scheme, like BDLOP [BDL⁺16] [LNP22, §1.3] or ABDLOP [LNP22], which is not succinct; it is important to stress that a succinct commitment scheme, like Ajtai's one, cannot be employed in our protocol since we need a compatibility between the commitments of ring elements and the commitments of their associated coefficients.
- (3) One might consider passing commitments to $f(\text{ct}(y'_{i,1}), \dots, \text{ct}(y'_{i,t}))$ directly in the above Protocol 6 however, this is likely not feasible. This is because the commitments we consider are lattice-based, and both the salts and the witnesses would end up with norms that are too large. Note that by considering R1CS structures and a small base b allows us to handle a reasonable amount of commitments, which would be exponential in the case of a general CCS structure.
- (4) The randomness δ_0 and δ_1 is sampled independently of γ , rather than being taken as further powers γ^K, γ^{2K+k} of the single batching challenge as in [NS26, Appendix B.2]. The reason is a soundness one, and it is the same reason the three $\tilde{\gamma}$ blocks are independent rather than powers of one another: item (2) of Lemma 4.16 needs the monomials of the three families to be distinguishable, and item (1) needs each block to enter with degree 1; cf. the proof of Lemma 4.9 and Proposition 4.17.
- (5) The attentive reader would have recalled that we are still committing to the instance-witness couple via $\mathcal{L}$ (recall Remark 2.23), thus, why is it necessary to consider different commitments? This is due to two (conflicting) reasons:
 - we necessitate of a non-succinct commitment scheme $\text{Commit}_{\text{ABDLOP}}$ (as noticed above), and
 - we wish to minimize the transmission and proof costs, therefore, any commitments which can be taken to be succinct must be so.
- (6) The anchoring salt in §3.2.1.1 ensures that we are binding T to the initial witness and that the final evaluations implies that Sum-Check has been correctly handled.
- (7) Passing $\sigma = \zeta P + T$ in plain discloses nothing about T . Indeed $P = \sum_{\mathbf{x} \in \{0,1\}^{\log_2 m}} p(\mathbf{x}) = 2^{\log_2 m} a_0 + (\cdot)$ is uniform in $\mathbb{K}$, since a_0 is uniform and $2^{\log_2 m} \not\equiv 0 \pmod{q}$ is invertible; as $\zeta \in \mathbb{K}^\times$, the product ζP is uniform as well, so σ is uniform regardless of T and T is perfectly hidden.
- (8) By considering the ABDLOP commitment scheme instead of a PCS, we replace the final evaluation check with a proof of knowledge, so that neither the masking polynomial's evaluation nor the real one is ever revealed. Explicitly, matching the notation of Protocol 5, the check certified is

$$h_{\log_2(m)}(r'_{\log_2(m)}) - \zeta \cdot p(\vec{r}') = Q(\vec{r}'),$$

where $h := h_{\log_2(m)}(r'_{\log_2(m)})$ is the prover's last-round Sum-Check message evaluated at the random point (public, exactly as in ordinary Sum-Check) and $p(\vec{r}')$ is the masking polynomial's value at $\vec{r}'$ (hidden, since p itself must stay hidden). This is precisely where we depart from [XZZ⁺19]: there the verifier obtains $g(\vec{r})$ from the opening of the polynomial commitment and recovers the real evaluation as $v_f = h - \rho g(\vec{r})$, which is the residual leak recorded in Remark 3.18.(1). Here $p(\vec{r}')$ is never opened, so $Q(\vec{r}')$ is never formed as a field element in $\mathcal{V}$'s view. The target $u^* := h - \text{eq}(r', \alpha)(c_F + \delta_0 c_N) - \delta_1 c_E$ is computed by $\mathcal{V}$ from quantities already in its view rather than transmitted, and, because c_F, c_N, c_E carry their salt aggregates (Step 16), it equals $\zeta p(\vec{r}')$ offset by the unopened salt aggregate of the Step 9–15 commitments. That offset is exactly what the weights $\{w_n\}$ of Step 18 account for, and it is what lets the final check be phrased as a single constant-coefficient linear relation in the committed messages, with a public right-hand side that $\mathcal{V}$ assembles from the transmitted BDLOP components (Lemma 3.10). The remaining public scalar of the masked rounds is h itself, which is harmless: see §4.1.1.1.

4.1.1.1. What $\mathcal{V}$ actually sees. Only three kinds of scalar reach $\mathcal{V}$ in the masked rounds. *Transmitted*: the Sum-Check round polynomials and hence the final value $h = (\zeta p + Q)(\vec{r}')$, the claimed sum $\sigma = \zeta P + T$, the commitments, and the challenges. *Derived by $\mathcal{V}$* , never transmitted: c_F, c_N, c_E (Step 16) and u^* (Step 17), which are deterministic functions of the former and so leak nothing further. *Never revealed*: the mask p and its evaluation $p(\vec{r}')$; the evaluations $Q(\vec{r}')$, $F(\vec{r}')$, $\text{NC}(\vec{r}')$, $\text{Eval}(\vec{r}')$; every hint $y'_{i,j}$ and surrogate $\hat{y}_{i,j}$

with its square, cube and product; and every salt. These are hidden by MLWE under freshly sampled salts, and are what $\mathcal{S}_{\text{ABDLOP}}$ and the simulator of Lemma 3.16 replace in hybrid $\mathbf{H}_1$ of Theorem 4.13 (including the invocations of Steps 1 and 17 of Protocol 7); for σ , see Remark 4.2.(7). Outside the masked rounds, $\mathcal{V}$ additionally holds u_1 itself. $\mathbf{x}_1$ is public by definition of the relation; c_1 is a deterministic function of $\mathbf{z}_1$, and over the committed relation it would be part of the instance and so handed to $\mathcal{S}$ outright. Over the unpadded relation of Definition 3.28 it is instead a prover message, and the blinded layout makes it statistically close to uniform (Lemma 3.3), which is what Theorem 4.13 exploits. The layout alone does not suffice here; see Remark 3.27. Without that layout the blinding guarantee is relative to c_1 , and the end-to-end claim would rest on the one-wayness rather than the hiding of $\mathcal{L}$ (however, without achieving zero-knowledge).

That leaves the transcript itself, where the mask gives more than a statistical-distance estimate would. Write $\ell = \log_2(m)$ and $d_i = \deg_{X_i} Q$. The mask $p = a_0 + \sum_i \tilde{p}_i(X_i)$ has $1 + \sum_i d_i$ uniform coefficients, and the consistent transcripts for degrees $(d_i)_{i \in [\ell]}$ form an affine space of the same dimension, each h_i contributing $d_i + 1$ coefficients constrained by $h_i(0) + h_i(1) = h_{i-1}(r'_{i-1})$. For fixed Q and $\zeta \in \mathbb{K}^\times$ the map between them is triangular, the coefficients of $\tilde{p}_i$ acting on the non-constant coefficients of h_i by $2^{\ell-i}$ in round i , and a_0 on the claimed sum by 2^ℓ , hence bijective for odd q . So for uniform p the whole transcript, h included, is uniform independently of Q : the masking is perfect, and it is unaffected by $\gamma, \delta_0, \delta_1, r'$ being public, since the randomness doing the work is p , which $\mathcal{V}$ never sees. The same computation gives $P = 2^\ell a_0 + 2^{\ell-1} \sum_{i,j} a_{i,j}$ uniform, whence the uniformity of $\zeta P + T$.

4.1.1.2. Soundness cost of the batched degree-0 check. The degree-0 surrogates $\{\hat{y}_{i,j}\}_{i \in [K+k], j \in [t]}$ of Protocol 6 (Step 10) are certified via a *single* $\Pi_{\text{many}}^{(1)}$ call for the vector of $N = 2(K+k)t$ linear relations, one row per (i, j) and per relation type. This vector-batching itself is free, as $\Pi_{\text{many}}^{(1)}$ natively proves $\mathbf{u} = \mathbf{R}_1 \mathbf{s}_1 + \mathbf{R}_m \mathbf{m} \in \mathbb{R}_{\mathbb{K}}^N$ for an arbitrary N , with no extra challenge and no soundness cost beyond $\Pi_{\text{many}}^{(1)}$'s own (cf. Lemma 3.6).

What is *not* free is collapsing, within each single $\hat{y}_{i,j}$, the $d-1$ scalar conditions $\text{ct}(\hat{y}_{i,j})_\ell = 0$, $\ell = 2, \dots, d$, into the one $\mathbb{R}_{\mathbb{K}}$ -valued row that $\Pi_{\text{many}}^{(1)}$ requires. We do this via the packaged rotation $\pi_\mu = \sum_{\ell=2}^d \mu_\ell \tau_\ell$ of Lemma 3.13, sampling a single $\{\mu_\ell\}_{\ell=2}^d \xleftarrow{\$} \mathbb{K}^{d-1}$ shared across all $(K+k)t$ surrogates. This is what lets N stay at $2(K+k)t$ rather than growing to $d(K+k)t$: without the collapse the first family would contribute $(d-1)(K+k)t$ rows and the second, whose members are one condition each, a further $(K+k)t$.

Each resulting row $\text{ct}(\pi_\mu \cdot \hat{y}_{i,j}) = 0$ is, however, only a constant-term statement, namely, $\pi_\mu \hat{y}_{i,j}$ is a nonzero ring element for an honest prover, so $\Pi_{\text{many}}^{(1)}$ cannot certify it directly (cf. §3.2.1). We route the whole family of $N = 2(K+k)t$ constant-term conditions through $\Pi_{\text{many}}^{(\text{ct})}$ (Protocol 4), which folds all of them into a single full-ring row under its own challenge vector $\vec{\gamma} \in \mathbb{K}^N$ before invoking $\Pi_{\text{many}}^{(1)}$, rather than certifying each row on its own.

This is where the actual soundness cost of batching sits. Fix a cheating prover with some $\hat{y}_{i_0, j_0}$ not degree-0. By Lemma 3.13 (applied to the single ring element $\hat{y}_{i_0, j_0}$), $\text{ct}(\pi_\mu \cdot \hat{y}_{i_0, j_0}) \neq 0$ except with probability at most $1/|\mathbb{K}|$ over $\{\mu_\ell\}$. Such bound is unaffected by how many surrogates are batched alongside it. Conditioned on that row being nonzero, Lemma 3.8 bounds the probability that $\Pi_{\text{many}}^{(\text{ct})}$ still accepts by $1/|\mathbb{K}|$ over $\vec{\gamma}$. Thus, the extracted identity is affine in the single coordinate γ_{j_0} attached to the offending row, with every other γ_j and the garbage term g held fixed, so this is a Schwartz–Zippel bound on one row rather than a union bound over N of them. Now, a union bound over these two (mutually independent) failure events gives a total knowledge-soundness cost for the batched degree-0 check of at most $\frac{2}{|\mathbb{K}|}$ (cf. Step 12 of Protocol 6).

4.1.1.3. The claimed sum σ (Protocol 6, Step 7). Here $N_c = 1 + kt$, indexed by $n \in \{P\} \cup (\{K+1, \dots, K+k\} \times [t])$, with

$$w_P := \zeta \cdot R_m^{(\Sigma)}, \quad w_{i,j} := \delta_1 \cdot \pi_{i,j}, \quad \pi_{i,j} := \sum_{\ell=1}^d \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)} \tau_\ell,$$

where $R_m^{(\Sigma)}$ is the packaged rotation realising $P = \sum_{\vec{x}} p(\vec{x})$ as a public linear functional of coeffs, and $\iota(n)$ is the row of B carrying coeffs resp. $y_{i,j}$. The public target of Definition 3.15 is simply $\mathbf{u} := \sigma$, and the certified relation is

$$\text{ct}\left(\zeta R_m^{(\Sigma)} \text{ coeffs} + \delta_1 \sum_{i=K+1}^{K+k} \sum_{j=1}^t \pi_{i,j} y_{i,j}\right) = \sigma.$$

Since this is already a statement about committed messages, Lemma 3.10 is not invoked at this call.

4.1.1.4. The target u^* (Protocol 6, Step 18). Here $N_c = 3K + 2k + kt + 1$, indexed by the four families

$$(4.1) \quad \{P\} \cup \{(i, \text{prod})\}_{i \in [K]} \cup \{(i, \text{cube}), (i, 1)\}_{i \in [K+k]} \cup \{(i, j)\}_{i \in [K+1, K+k], j \in [t]},$$

of sizes 1, K , $2(K+k)$ and kt respectively, with

$$(4.2) \quad w_P := \zeta \cdot R_m^{(r')}, \quad w_{i, \text{prod}} := \text{eq}(r', \alpha) \gamma_i^{(1)}, \quad w_{i, \text{cube}} := \delta_0 \text{eq}(r', \alpha) \gamma_i^{(1)},$$

$$(4.3) \quad w_{i,1} := -(\mathbf{1}_{i \leq K} + \delta_0) \text{eq}(r', \alpha) \gamma_i^{(1)}, \quad w_{i,j} := \delta_1 \text{eq}(r', r) \pi_{i,j},$$

for $R_m^{(r')}$ the packaged rotation sending **coeffs** to the constant slot of $p(\vec{r}')$. The slot subtracted by the R1CS predicate $f(Y_1, Y_2, Y_3) = Y_2 Y_3 - Y_1$ is $j = 1$, because $\mathbf{M}_1 = \mathbf{I}_m$ (Protocol 6, parameters), and it is the same slot that NC consumes. The two families therefore reach the *same* surrogate $\hat{y}_{i,1}$ for $i \in [K]$, and their weights are summed there, which is what the indicator $\mathbf{1}_{i \leq K}$ records. The four families are disjoint. In particular the $(i, 1)$ -blocks and the (i, j) -blocks are *not* identified for $i > K$. The former carry the surrogate salts $\hat{s}_{2,i,1}$ of Step 10 while the latter the hint salts $\mathbf{s}'_{2,i,1}$ of Step 9, and these are independently sampled. Step 12 certifies $\text{ct}(y'_{i,1}) = \text{ct}(\hat{y}_{i,1})$, an equality of committed *values*. It says nothing about the salts, and the two objects cannot be merged; cf. Step 10 of Protocol 6. Where a single *commitment* is reached by several weights (which happens whenever several of the messages above are packed into distinct BDLOP slots of one commitment) that commitment contributes one block, its weights being summed into its column group of $\mathbf{R}_m$. Under the maximal packing described there the count becomes

$$(4.4) \quad N_c^{(18)} = 1 + \left\lceil \frac{2(K+k)+K}{\ell_{\text{aux}}} \right\rceil + \left\lceil \frac{(K+k)t}{\ell_{\text{aux}}} \right\rceil + k,$$

ℓ_{aux} being the number of BDLOP message slots used for the never-folded objects.

With $u^* := h - \text{eq}(r', \alpha)(c_F + \delta_0 c_N) - \delta_1 c_E$ computed by $\mathcal{V}$ as above (a public value), the anchoring call with these weights certifies $u^* = \zeta \cdot S_{2,p} = \text{ct}(\sum_n w_n B \mathbf{s}_2^{(n)})$, i.e. that the public u^* equals ζ times the salt contribution underlying the single committed quantity $p(\vec{r}')$; equivalently, by Lemma 3.10, that $\text{ct}(\sum_n w_n \mathbf{m}_{\iota(n)}^{(n)}) = \text{ct}(\mathbf{u})$ for the public $\mathbf{u}$ assembled from the transmitted $\mathbf{t}_B^{(n)}$. Combined with the definition of u^* , this is exactly the identity $h - \zeta p(\vec{r}') = Q(\vec{r}')$ certified at the level of the commitments.

4.1.1.5. A remark on soundness of Protocol 6 as a whole. Anchoring σ and the verifier-computed u^* via Π^{anc} is what lets these checks be knowledge-sound rather than merely a formal identity a cheating prover could satisfy with an arbitrary field value. Lemma 3.16 guarantees that whatever σ and u^* the verifier accepts are provably the asserted functions of the same $(\mathbf{m}^{(n)}, \mathbf{s}_2^{(n)})$ underlying the commitments opened earlier in the protocol. This is exactly the extraction hook used in the proof of Lemma 4.9, and together with the ordinary ZKPoK is what lets the extractor of Protocol 14 pull out consistent salts $\{s_{i,j}\}$.

4.1.2. **Second Reduction: Random Linear Combination.** Besides the commitment modifications, we must guarantee that the distribution of the random linear combination can be simulated. To this goal, we employ the results recalled in §3.6. Note that the algorithm still maintains its PPT structure. We highlight such modification in **brown**.

Protocol 7 Random linear combination reduction Π'_{RLC}

Parameters: Refer to Definition 2.2. **Maximal repetition counter:** W_{max} .

Input: $\in \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^{K+k}$

$(s; c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\text{F}, \text{in}}}, r \in \mathbb{K}^{\log_2(m)}, (\{\text{Commit}(y_{i,j}, \mathbf{s}_{i,j}) \in \mathbb{R}_{\mathbb{F}}^{\kappa_{\text{com}}}\}_{j \in [t]}; \{y_{i,j} \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, \{\mathbf{s}_{i,j} \in \mathbb{R}_{\mathbb{F}}^{m_1} \times \mathbb{R}_{\mathbb{F}}^{m_2}\}_{j \in [t]}, z_i \in \mathbb{F}^{n_{\text{F}}})_{i \in [K+k]}$

Output: $\in \text{CE}_{\text{com}}(B, \mathcal{L}, B_{\text{fold}}, \text{Commit}) \subseteq \text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit})$ $\triangleright$ Lemma B.1 for the bound, Remark 3.23 for the inclusion; soundness is stated against the relaxed relation.

$(s; c \in \mathbb{C}, x \in \mathbb{F}^{n_{\text{F}, \text{in}}}, r \in \mathbb{K}^{\log_2(m)}, \{\text{Commit}(y_j, \mathbf{s}_j) \in \mathbb{R}_{\mathbb{F}}^{\kappa_{\text{com}}}\}_{j \in [t]}; \{y_j \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, \{\mathbf{s}_j \in \mathbb{R}_{\mathbb{F}}^{m_1} \times \mathbb{R}_{\mathbb{F}}^{m_2}\}_{j \in [t]}, z \in \mathbb{F}^{n_{\text{F}}})$

Setup $\mathcal{G}(1^\lambda, n_{\mathbb{R}}) \rightarrow \text{pp}$: Output $\text{pp} \leftarrow \text{Setup}(1^\lambda, n_{\mathbb{R}})$.

Encoder $\mathcal{K}(\text{pp}, s) \rightarrow (\text{pk}, \text{vk})$: Output $((\text{pp}, s), \perp)$.

Reduction $\langle \mathcal{P}, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, w_1) \rightarrow (u_2, w_2)$:

- 1: $\mathcal{V} \leftrightarrow \mathcal{P}$: For each $j \in [t]$ and each $i \in [K+k]$, the prover and the verifier engage in the proof of knowledge for the commitments $y_{i,j} \in \mathbb{R}_{\mathbb{K}}$ as in Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$; cf. also §2.5.2.
- 2: **while** TRUE **do**
- 3: $\mathcal{P}$: Sample masking value $\mathbf{y} \leftarrow D_{\mathbf{s}}^{n_{\mathbb{F}}}$. ▷ Independent of every other challenge and sampled value.
- 4: $\mathcal{P}$: Compute $c_{\mathbf{y}} := \mathcal{L}(\mathbf{y})$ and $\mathbf{x}_{\mathbf{y}} := \mathcal{L}_{\text{in}}(\mathbf{y})$, and $\forall j \in [t]$, $y_{\mathbf{y},j} := \widetilde{\mathbf{M}}_j \mathbf{y}(r)$.
- 5: $\mathcal{P}$: Sample fresh salts $\mathbf{s}_{1,\mathbf{y},j}, \mathbf{s}_{2,\mathbf{y},j}$ and commit $C_{\mathbf{y},j} := \text{Commit}(y_{\mathbf{y},j}, \mathbf{s}_{1,\mathbf{y},j}, \mathbf{s}_{2,\mathbf{y},j})$, $\forall j \in [t]$.
- 6: $\mathcal{P}$: Sample a further, *independent* salt pair $(\mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$ with $\|\mathbf{s}_{l,\mathbf{y},0}\|_{\infty} < \tilde{B}$, and commit to the pair $C_{\mathbf{y},0} := \text{Commit}((c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}), \mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$. ▷ The entries have unbounded norm, so they occupy BDLOP slots, and together they overflow one commitment: $C_{\mathbf{y},0}$ denotes a tuple of $\tilde{N} = \lceil (\kappa + n_{\text{R,in}})/\ell \rceil$ commitments at the ambient (ℓ, m_2) , one per block of the message, sent and opened together (Remark 4.21). The salt block is disjoint from those of $\{C_{\mathbf{y},j}\}$, as $C_{\mathbf{y},0}$ alone is opened in the clear; both values are bound here but disclosed only once the loop breaks. Cf. Remarks B.3 and B.9.
- 7: $\mathcal{P} \rightarrow \mathcal{V}$: Send $C_{\mathbf{y},0}$ and $\{C_{\mathbf{y},j}\}_{j \in [t]}$.
- 8: $\mathcal{V}$: Increment the attempt counter w .
If $w > W_{\text{max}}$, output $\perp$ and halt. Otherwise sample $\rho_1, \dots, \rho_{K+k} \xleftarrow{\$} \mathcal{C}$ and send them to $\mathcal{P}$. ▷ Cf. Remark 4.3 and §2.1.
- 9: $\mathcal{P}$: Compute $\mathbf{v} \leftarrow \sum_{i \in [K+k]} \rho_i \mathbf{z}_i$.
- 10: $\mathcal{P}$: Compute $\mathbf{z} \leftarrow \mathbf{v} + \mathbf{y}$.
- 11: **if** $\text{Rej}_1(\mathbf{z}, \mathbf{v}, \mathbf{s})$ **accepts** **then**
- 12: $\mathcal{P} \rightarrow \mathcal{V}$: Send the opening $((c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}), \mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$ of $C_{\mathbf{y},0}$.
- 13: $\mathcal{V}$: Check $C_{\mathbf{y},0} \stackrel{?}{=} \text{Commit}((c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}), \mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$ **and** $\|\mathbf{s}_{l,\mathbf{y},0}\|_{\infty} < \tilde{B}$ for $l \in [2]$; abort otherwise.
▷ Both checks are in the clear: the salts being revealed, the norm is read off coefficient-wise at cost $\mathcal{O}((m_1 + m_2)d)$. Cf. Remark B.9.
- 14: **Break.**
- 15: **end if**
- 16: **end while**
- 17: $\mathcal{V} \leftrightarrow \mathcal{P}$: For each $j \in [t]$, the prover and the verifier engage in the proof of knowledge for the commitment $y_{\mathbf{y},j} \in \mathbb{R}_{\mathbb{K}}$, as in Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$.
- 18: $\mathcal{V}$: Compute:

$$c \leftarrow \sum_{i \in [K+k]} \rho_i c_i + c_{\mathbf{y}}, \quad \mathbf{x} \leftarrow \sum_{i \in [K+k]} \rho_i \mathbf{x}_i + \mathbf{x}_{\mathbf{y}}.$$
- 19: $\mathcal{P}$: Compute $s_{1,j} = \sum_i \rho_i s_{1,i,j} + \mathbf{s}_{1,\mathbf{y},j}$ and $s_{2,j} = \sum_i \rho_i s_{2,i,j} + \mathbf{s}_{2,\mathbf{y},j}$.
- 20: $\mathcal{P} \rightarrow \mathcal{V}$: Compute $\forall j \in [t]$, $y_j \leftarrow \sum_{i \in [K+k]} \rho_i y_{i,j} + y_{\mathbf{y},j}$, commit as $\text{Commit}(y_j)$ (w.r.t. $s_{1,j}, s_{2,j}$), then send.
- 21: $\mathcal{V}$: For each $j \in [t]$, check $\text{Commit}(y_j) \stackrel{?}{=} \sum_{i \in [K+k]} \rho_i \text{Commit}(y_{i,j}) + C_{\mathbf{y},j}$.
- 22: Output $(\mathbf{s}; c, x, r, \{\text{Commit}(y_j)\}_{j \in [t]}; \{y_j\}_{j \in [t]}, z)$.

The mask $\mathbf{y}$, and the quantities $c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}, y_{\mathbf{y},j}$ derived from it, are never scaled by any ρ_i : they are fixed once, before $\rho_1, \dots, \rho_{K+k}$ are drawn (and only ever resampled together, as a block, on a Rej_1 rejection), and enter $c, \mathbf{x}, y_j$ additively. In particular $\mathbf{y}$ never occupies one of the $K+k$ challenge-indexed slots.

The mask $\mathbf{y}$ must precede the challenge. Lemma 3.20 is a statement about a specific sampling procedure: $\mathbf{y} \leftarrow D_{\mathbf{s}}^{n_{\mathbb{F}}}$ drawn independently of the secret v it is meant to hide, so that a simulator can reproduce the marginal of $\mathbf{z} = v + \mathbf{y}$ without ever knowing v . In Protocol 7, $v := \sum_{i \in [k+K]} \rho_i \mathbf{z}_i$ depends on the verifier's own challenge $\rho_{K+1}, \dots, \rho_{K+k}$, so it cannot itself play the role of the lemma's independent mask. Conversely, $\mathbf{y}$ is sampled by the prover before $\rho_1, \dots, \rho_{K+k}$ are even drawn, and added unscaled by any ρ_i . This is what lets us invoke Lemma 3.20 honestly, with v playing the role of the arbitrary secret and $\mathbf{y}$ the role of the genuine Gaussian mask.

Remark 4.3 (On the Budget Attempt). Recall that (§2.1) $W_{\text{max}} := \left\lceil \lambda / \log_2(M/(M-1)) \right\rceil$ is the attempt budget, where M is the repetition rate of the rejection sampling function Rej_1 . W_{max} is public and known to both parties, and $\mathcal{V}$ can enforce it because it samples $\vec{p}$ afresh on every iteration. Enforcement costs one bit per attempt ($\mathcal{P}$ signals accept or restart, the two cases being distinguished by whether an opening or a fresh commitment follows) but no additional round. An honest $\mathcal{P}$ exhausts the budget with probability

at most $(1 - 1/M)^{W_{\max}} \leq 2^{-\lambda}$; this forces our protocols to be statistically complete rather than achieving perfect completeness. Let us remark that this bound is what makes the extraction argument of Lemma 4.10 go through; see Remark B.2. At the working level $\lambda = 120$ fixed in §2.1 this is $W_{\max} = 206$ for $M \approx 3$ and $W_{\max} = 120$ for $M \approx 2$ (for $\lambda = 128$, $W_{\max} = 219, 128$ respectively). Unlike M , which governs the expected number of attempts, $W_{\max}$ is a worst-case bound enforced by $\mathcal{V}$ and is the quantity that enters the knowledge-soundness error; cf. Remark B.2. See Proposition 4.15.

4.1.2.1. Batching convention.

Remark 4.4 (Batching the Π'_{RLC} proofs of knowledge). Step 1 of Protocol 7 is stated as $(K + k)t$ separate invocations, one per commitment, purely for readability. They can be collapsed into a single call to $\Pi_{\text{many}}^{(1)}$ on the concatenation of the $(K + k)t$ salt blocks, exactly as Step 5 of Protocol 8 already does on $(k + 1)t$ of them, and likewise the t invocations of Step 17. Nothing in the analysis changes: $\Pi_{\text{many}}^{(1)}$ natively proves $\mathbf{u} = \mathbf{R}_1 \mathbf{s}_1 + \mathbf{R}_m \mathbf{m} \in \mathbb{R}_{\mathbb{K}}^N$ for arbitrary N , and the vector-batching is free both in soundness (§4.1.1.2) and, now, in completeness, since one rejection loop replaces $(K + k + 1)t$ of them.

The gain is worth recording, because this term dominates the completeness error of the whole protocol. Batched, $\varepsilon_c^{\text{RLC}}$ drops from $(1 + (K + k + 1)t)2^{-\lambda}$ to $3 \cdot 2^{-\lambda}$ and the consolidated $\varepsilon_c^{\text{LBF}}$ of Proposition 4.15 scales accordingly.

It is worth fixing, once and for all, the convention under which every completeness count in this paper is stated, since it is the only source of asymmetry between the three reductions.

Our protocols are written with the ABDLOP proofs of knowledge spelled out invocation by invocation, one per commitment, because that is the form in which the extraction arguments of Lemmas 4.9–4.12 are easiest to read: each invocation certifies exactly one salt block, and the extractor recovers exactly one opening from it. Step 1 of Protocol 6 and Step 5 of Protocol 8 are displayed already batched, whereas Steps 1 and 17 of Protocol 7 are not. This asymmetry is an accident of exposition and not one of substance. By Remark 4.4 the latter two collapse into two $\Pi_{\text{many}}^{(1)}$ calls at no cost in soundness, in communication, or in the extractor, since $\Pi_{\text{many}}^{(1)}$ natively proves $\mathbf{u} = \mathbf{R}_1 \mathbf{s}_1 + \mathbf{R}_m \mathbf{m} \in \mathbb{R}_{\mathbb{K}}^N$ for arbitrary N .

Accordingly, the batched form is the intended implementation path, and every count of the shape $N_{\text{PoK}} = 6 + (K + k + 1)t$, together with every completeness error derived from it, is reported as a conservative upper bound that can be read directly off the displayed protocols. Under the default batched path the same statements read

$$(4.5) \quad N_{\text{PoK}} = 8, \quad \varepsilon_c^{\text{RLC}} \leq 3 \cdot 2^{-\lambda}, \quad \varepsilon_c^{\text{LBF}} \leq 9 \cdot 2^{-\lambda},$$

uniformly in k and t , and deduplicating as in Remark 4.14 brings N_{PoK} further down to 7.

4.1.3. Third Reduction: Decomposition Reduction. Recall the split_b algorithm for computing a b -ary decomposition of a norm-bounded $\mathbf{z}$ ($\|\mathbf{z}\|_{\infty} < b^k$) as $\mathbf{z} = \sum_{i=1}^k b^{i-1} \mathbf{z}_i$ with $\|\mathbf{z}_i\|_{\infty} < b$. Only the witness is decomposed while the commitment salts are freshly sampled. The consistency of the pieces with the input commitment is certified by a batched proof of knowledge rather than by a homomorphic identity.

Protocol 8 Decomposition reduction Π'_{DEC} (fresh salts, commit-and-prove)

Parameters: Refer to Definition 2.2.

Input: $\in \text{CE}_{\text{com}}(B, \mathcal{L}, B_{\text{fold}}, \text{Commit}) \subseteq \text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit}) \triangleright$ tight for completeness, relaxed for soundness; Remark 3.23

$(\mathbf{s}; c \in \mathbb{C}, x \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}, r \in \mathbb{K}^{\log_2(m)}, \{C_j := \text{Commit}(y_j, \mathbf{s}_j) \in \mathbb{R}_{\mathbb{F}}^{\kappa_{\text{com}}}\}_{j \in [t]}; \{y_j \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, \{\mathbf{s}_{l,j}\}_{(l,j) \in [2] \times [t]}, z \in \mathbb{F}^{n_{\mathbb{F}}})$

Output: $\in \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$

$(\mathbf{s}; c_i, x_i, r, \{\text{Commit}(y_{i,j}, \mathbf{s}_{i,j})\}_{j \in [t]}; \{y_{i,j}\}_{j \in [t]}, \{\mathbf{s}_{l,i,j}\}_{(l,j) \in [2] \times [t]}, z_i)_{i \in [k]}$

Setup $\mathcal{G}(1^\lambda, n_{\mathbb{R}}) \rightarrow \text{pp}$: Output $\text{pp} \leftarrow \text{Setup}(1^\lambda, n_{\mathbb{R}})$.

Encoder $\mathcal{K}(\text{pp}, \mathbf{s}) \rightarrow (\text{pk}, \text{vk})$: Output $((\text{pp}, \mathbf{s}), \perp)$.

Reduction $\langle \mathcal{P}, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, w_1) \rightarrow (u_2, w_2)$:

- 1: $\mathcal{P}$: Compute $(z_1, \dots, z_k) \leftarrow \text{split}_b(z)$ and, for $i \in [k]$,

- $c_i \leftarrow \mathcal{L}(\mathbf{z}_i)$, $y_{i,j} \leftarrow \widetilde{M_j z_i}(r)$ for $j \in [t]$.
- 2: $\mathcal{P}$: For every $i \in [k]$, $j \in [t]$ sample a *fresh* salt pair $\mathbf{s}_{i,j} := (\mathbf{s}_{1,i,j}, \mathbf{s}_{2,i,j})$, uniform subject to $\|\mathbf{s}_{l,i,j}\|_\infty < \tilde{B}$, and set $C_{i,j} := \text{Commit}(y_{i,j}, \mathbf{s}_{i,j})$. ▷ Cf. Remark 4.5.
- 3: $\mathcal{P} \rightarrow \mathcal{V}$: Send $(c_i, \{C_{i,j}\}_{j \in [t]})_{i \in [k]}$ to $\mathcal{V}$.
- 4: $\mathcal{V}$: Compute $(x_1, \dots, x_k) \leftarrow \text{split}_b(x)$ and check
- $$c \stackrel{?}{=} \sum_{i \in [k]} b^{i-1} \cdot c_i, \quad \text{where the norm bound } b \text{ is treated as a field element.}$$
- 5: $\mathcal{V} \leftrightarrow \mathcal{P}$: the parties run a single $\Pi_{\text{many}}^{(1)}$ (Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$) on the concatenation of $(C_j, \{C_{i,j}\}_{i \in [k]})_{j \in [t]}$, i.e. $N_c = (k+1)t$ salt blocks, with $\mathbf{u} = 0 \in \mathbb{R}_{\mathbb{K}}^t$, $\mathbf{R}_1 = 0$, and $\mathbf{R}_m$ the $t \times N_c$ block-row matrix whose j -th row is $(1, -1, -b, \dots, -b^{k-1})$ on the j -th block, at width $\mathbf{s}_2$ calibrated to B_{fold} , certifying
- $$y_j - \sum_{i \in [k]} b^{i-1} y_{i,j} = 0 \quad \text{for all } j \in [t].$$
- ▷ Replaces the homomorphic check $C_j \stackrel{?}{=} \sum_i b^{i-1} C_{i,j}$. The message identity is exact by Lemma 3.6(3); the salt bounds are the relaxed ones of item (2), which is all one invocation of Π_{LBF} needs. Cf. Remark 4.8.
- 6: $\mathcal{V}$: accept iff Steps 4 and 5 accept. Output $(\mathbf{s}; c_i, x_i, r, \{C_{i,j}\}_{j \in [t]}; \{y_{i,j}\}_{j \in [t]}, \{\mathbf{s}_{l,i,j}\}, z_i)_{i \in [k]}$
-

Remark 4.5 (Why the salts are resampled rather than decomposed). *Decomposing the salt might be degenerate. Lemma B.1 bounds a folded salt by $B_{\text{fold}} = (K+k)T(\mathcal{C})(\tilde{B}-1) + \tilde{B}$, whereas a $\text{split}_{\tilde{B}}$ into k pieces inherits k from the witness budget $\tau_{\lambda,\xi} \mathbf{s} < b^k$ of §3.6. The two budgets differ by exactly the rejection-sampling blow-up*

$$\frac{\tau_{\lambda,\xi} \xi (K+k) T(\mathcal{C}) (b-1) \sqrt{n_{\mathbb{F}}}}{(K+k) T(\mathcal{C}) (\tilde{B}-1)} = \tau_{\lambda,\xi} \xi \sqrt{n_{\mathbb{F}}} \approx 2^{17.3}$$

at $n_{\mathbb{F}} = 2^{21}$, $\xi \approx 20$, $\tau_{\lambda,\xi} \approx 5.60$ (the $\lambda = 120$ value of §3.6), so that ≈ 17 of the $k \in \{30, 31\}$ digits are identically zero. For those indices $\mathbf{s}_{1,i,j} = \mathbf{s}_{2,i,j} = 0$ and the ABDLOP commitment degenerates and the evaluation hint would be transmitted in the clear. Similarly, Lemma 3.6 would not apply to those commitments, since $\mathcal{S}_{\text{ABDLOP}}$ samples (t_A, t_B) uniformly and no zero-salt commitment resembles a uniform pair.

4.2. The LatticeBlindFold protocol. Before providing the precise construction of our LatticeBlindFold protocol, we need to define a sampling algorithm for randomized instance-witness pairs in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$.

Protocol 9 Sample a randomized $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ instance-witness pair.

Algorithm: $(\mathbf{u}, \mathbf{w}) \leftarrow \text{sample}_{\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})}(\mathbf{pp}, \mathbf{s})$.

Input: Public parameters $\mathbf{pp}$ and a $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ structure $\mathbf{s}$.

Output: A randomized $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ instance-witness pair $(\mathbf{u}, \mathbf{w})$.

- 1: Parse size bounds $t, n_{\mathbb{R}}, n_{\mathbb{R}, \text{in}}, m, b, \tilde{B}$, and $\kappa_{\text{com}} = \kappa + \kappa'$ from $\mathbf{s}$ and $\mathbf{pp}$.
 - 2: Parse matrices $M_1, \dots, M_t$ from the $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ structure $\mathbf{s}$.
 - 3: Randomly sample partial instances $(x, r) \in (\mathbb{F}^{n_{\mathbb{R}, \text{in}}}, \mathbb{K}^{\log_2(m)})$ and witnesses $w \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{R}, \text{in}}}$, whose last $n_{\mathbb{F}, \text{bl}}$ coordinates form the blinding block of Definition 3.1, such that $z := [x, w] \in \mathbb{F}^{n_{\mathbb{F}}}$ has norm $\|z\|_\infty < b$.
 - 4: Compute $c \leftarrow \mathcal{L}(\mathbf{z})$.
 - 5: For $j \in [t]$ set $y_j \leftarrow \widetilde{M_j \mathbf{z}}(r)$.
 - 6: For $j \in [t]$ and $l \in [2]$, randomly sample $\mathbf{s}_{l,j} \in \mathbb{R}_{\mathbb{F}}^{m_l}$ with $\|\mathbf{s}_{l,j}\|_\infty < \tilde{B}$, write $\mathbf{s}_j := (\mathbf{s}_{1,j}, \mathbf{s}_{2,j})$, and compute $C_j \leftarrow \text{Commit}(y_j, \mathbf{s}_j)$.
 - 7: Compute $\mathbf{u} \leftarrow (c, x, r, (C_1, \dots, C_t))$ and $\mathbf{w} \leftarrow (z, (y_1, \dots, y_t), (\mathbf{s}_1, \dots, \mathbf{s}_t))$.
 - 8: **return** $(\mathbf{u}, \mathbf{w})$.
-

Protocol 10 Sample a randomized $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ instance-witness pair.

Algorithm: $(u, w) \leftarrow \text{sample}_{\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k}(\text{pp}, s)$.

Input: Public parameters pp and a $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ structure s .

Output: A randomized $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ instance-witness pair (u, w) .

- 1: Compute $(u_i, w_i) \leftarrow \text{sample}_{\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})}(\text{pp}, s)$ for $i \in [k]$. $\triangleright$ Here we can parse sizes and structures only once.
 - 2: **return** $(u_i, w_i)_{i \in [k]}$.
-

In the end we provide a (PPT) sampling algorithm which samples a $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ instance-witness pair where the distribution of the witness z is indistinguishable from a genuine output of the LatticeBlindFold protocol (Protocol 12), that is, an instance-witness pair in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$ (Definition 3.25).

Protocol 11 Sample a randomized $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$ instance-witness pair.

Algorithm: $(u, w) \leftarrow \text{sample}_{\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k}^{\text{blind}}(\text{pp}, s)$.

Input: Public parameters pp and a $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ structure s .

Output: A randomized $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ instance-witness pair (u, w) where $\sum_{i \in [k]} z_i$ distributes as a discrete Gaussian $D_{\mathbb{S}}^{n_{\mathbb{F}}}$; cf. §3.6.

- 1: Parse size bounds $t, n_{\mathbb{R}}, n_{\mathbb{R}, \text{in}}, m, b, \tilde{B}$, and $\kappa_{\text{com}} = \kappa + \kappa'$ from s and pp .
 - 2: Parse matrices $M_1, \dots, M_t$ from the $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})$ structure s .
 - 3: Randomly sample partial instances $r \leftarrow \mathbb{K}^{\log_2(m)}$
 - 4: Randomly sample $z := [x, w] \in \mathbb{F}^{n_{\mathbb{F}}}$ as $z \leftarrow D_{\mathbb{S}}^{n_{\mathbb{F}}}$, restarting while $\|z\|_{\infty} > \tau_{\lambda, \xi} \cdot s$. $\triangleright$ Up to negligible probability $\|z\| \leq \tau_{\lambda, \xi} \xi (K + k)T(b - 1)\sqrt{n_{\mathbb{F}}} < B = b^k$; cf. §3.6.
 - 5: Compute the decomposition $(z_1, \dots, z_k) \leftarrow \text{split}_b(z)$.
 - 6: Compute $x_i \leftarrow \mathcal{L}_{\text{in}}(\mathbf{z}_i)$ and $c_i \leftarrow \mathcal{L}(\mathbf{z}_i)$, for $i \in [k]$.
 - 7: For $j \in [t]$ set $y_{i,j} \leftarrow \widehat{M_j \mathbf{z}_i}(r)$.
 - 8: For $i \in [k], j \in [t]$ and $l \in [2]$, sample independent salts $\mathbf{s}_{l,i,j} \in \mathbb{R}_{\mathbb{F}}^{m_l}$, uniform subject to $\|\mathbf{s}_{l,i,j}\|_{\infty} < \tilde{B}$, and write $\mathbf{s}_{i,j} := (\mathbf{s}_{1,i,j}, \mathbf{s}_{2,i,j})$.
 - 9: Compute $C_{i,j} \leftarrow \text{Commit}(y_{i,j}, \mathbf{s}_{i,j})$.
 - 10: Set $u \leftarrow (c_i, x_i, r, (C_{i,1}, \dots, C_{i,t}))_{i \in [k]}$ and $w \leftarrow (\mathbf{z}_i, (y_{i,1}, \dots, y_{i,t}), (\mathbf{s}_{i,1}, \dots, \mathbf{s}_{i,t}))_{i \in [k]}$.
 - 11: **return** (u, w) .
-

We can finally combine the above Protocols 6, 8, and 7. We define the LatticeBlindFold protocol as it follows. Recall that $K = 1$ and the definitions in §3.7 and we assume here that the pre-composition with ι_{bl} is in place, performing the embedding as its first step.

Protocol 12 The LatticeBlindFold protocol Π_{LBF} .

Fold K R1CS instance-witness pairs and introduce zero-knowledge on top of the given R1CS circuit.

Input: $\in \text{CCS}(b, \mathcal{L})^K$

$(s; (x_i \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}; w_i \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}, \text{in}}})_{i=1}^K)$.

Output: $\in \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k \subset \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$

$(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, \mathbf{s}'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, \{z'_i\}_{i \in [k]})$

Precondition: $K = 1$. $\mathcal{V}$ rejects immediately if the input tuple's length does not match $K = 1$; the blinding guarantee of Theorem 4.13 is proved only for this case.

- 1: $\mathcal{P}$: The prover samples an instance-witness pair in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$ (Protocol 11):
 $(s; c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}, \text{in}}}, r \in \mathbb{K}^{\log_2(m)}; \{y_{i,j} \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, z_i \in \mathbb{F}^{n_{\mathbb{F}}})_{i=K+1}^{K+k} \leftarrow \text{sample}_{\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K=1)^k}(\text{pp}, s)$.

- 2: $\mathcal{P} \leftrightarrow \mathcal{V}$: Prover and verifier engage in the reduction $\Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$ (in application order, Protocols 6,7,8) for the pair

$$\left((s; (c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}); w_i \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}}, \text{in}})_{i=1}^K, \right. \\ \left. (s; c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}, r \in \mathbb{K}^{\log_2(m)}; \{y_{i,j} \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, z_i \in \mathbb{F}^{n_{\mathbb{F}}})_{i=K+1}^{K+k} \right)$$

where $c_i \in \mathbb{C}$, $i \in [K]$ are obtained by applying the compact Ajtai commitment $\mathcal{L}$ (cf. Equation 3.18).

The procedure outputs the randomized tuple $(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, s'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, z'_i)_{i \in [k]}$.

- 3: **return** $(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, s'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, z'_i)_{i \in [k]}$.

Definition 4.6. By abuse of notation, we define the *LatticeBlindFold* protocol as the composition of the above Protocol 12 with the map ι_{bl} of Equation 3.18, where we add a first prover step: the prover engages in the randomisation of Equation (3.19) and publish the Ajtai commitments c_i , $i \in [K]$. The *LatticeBlindFold* Π_{LBF} protocol then takes inputs in $\text{CCS}^\circ(b)^K \subset \text{CCS}(b, \mathcal{L})^K$ (Definition 3.28).

4.2.1. A blueprint for LatticeBlindFold as a zero-knowledge folding scheme. As stated, achieving a full folding scheme is out of the scope of our work. However, we can hint to how we can employ the *LatticeBlindFold* step Π_{LBF} to obtain a zero-knowledge folding scheme, which we denote by $\Pi_{\text{LBF-Fold}}$. As above we have $K = 1$ (thus, we would have a folding scheme, not a multi-folding scheme).

Protocol 13 The *LatticeBlindFold* Blueprint for a Folding Scheme $\Pi_{\text{LBF-Fold}}$.
Fold K R1CS instance-witness pairs in a zero-knowledge fashion.

Input: $\in \text{CCS}^\circ(b)^K \subset \text{CCS}(b, \mathcal{L})^K$ (Definition 3.28)

$$(s; (x_i \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}); w_i \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}}, \text{in}})_{i=1}^K.$$

Optional Input: $\in \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k \subset \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$

$$(s; c_i, x_i, r, \{\text{Commit}(y_{i,j}, s_{i,j})\}_{j \in [t]}; \{y_{i,j}\}_{j \in [t]}, z_i)_{i=K+1}^{K+k}$$

Output: $\in \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$

$$(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, s'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, z'_i)_{i \in [k]}$$

Precondition: $K = 1$. $\mathcal{V}$ rejects immediately if the input tuple's length does not match $K = 1$; the blinding guarantee of Theorem 4.13 is proved only for this case.

- 1: **if** **Optional Input** $= \perp$ **then**

- 2: $\mathcal{P} \leftrightarrow \mathcal{V}$: Prover and verifier engage in Π_{LBF} (Protocol 12) on input

$$(s; (c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}); w_i \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}}, \text{in}})_{i=1}^K$$

obtaining the randomized tuple $(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, s'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, z'_i)_{i \in [k]}$.

- 3: **else if** **Optional Input** $= (s; c_i, x_i, r, \{\text{Commit}(y_{i,j}, s_{i,j})\}_{j \in [t]}; \{y_{i,j}\}_{j \in [t]}, z_i)_{i=K+1}^{K+k}$ **then**

- 4: $\mathcal{P} \leftrightarrow \mathcal{V}$: Prover and verifier engage in the reduction $\Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$ (in application order, Protocols 6,7,8) for the pair

$$\left((s; (c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}); w_i \in \mathbb{F}^{n_{\mathbb{F}} - n_{\mathbb{F}}, \text{in}})_{i=1}^K, \right. \\ \left. (s; c_i \in \mathbb{C}, x_i \in \mathbb{F}^{n_{\mathbb{F}}, \text{in}}, r \in \mathbb{K}^{\log_2(m)}; \{y_{i,j} \in \mathbb{R}_{\mathbb{K}}\}_{j \in [t]}, z_i \in \mathbb{F}^{n_{\mathbb{F}}})_{i=K+1}^{K+k} \right)$$

where $c_i \in \mathbb{C}$, $i \in [K]$ are obtained by applying the compact Ajtai commitment $\mathcal{L}$ (cf. Equation 3.18).

The procedure outputs the randomized tuple $(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, s'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, z'_i)_{i \in [k]}$.

- 5: **end if**

- 6: **return** $(s; c'_i, x'_i, r', \{\text{Commit}(y'_{i,j}, s'_{i,j})\}_{j \in [t]}; \{y'_{i,j}\}_{j \in [t]}, z'_i)_{i \in [k]}$.

It is crucial that the blinding step is run at first, otherwise the final Sum-Check evaluation would break zero-knowledge. We addressed the branch of Protocol 13 in which the optional input is $\perp$ and the prover samples the k accumulator instances itself via Protocol 11. In the other branch the accumulator is supplied from outside, no fresh sampling occurs, and nothing above applies to it. We stress again that this is an informal design principle, not a proof. Formalizing it requires a *preservation* statement, of the shape: if

the input accumulator is distributed as $\text{sample}^{\text{blind}}$ then so is the output. Such statement must be proved inductively over rounds. We have not established it, and we flag two places where it is not free. First, the mask of Protocol 7 is drawn afresh each round, so the per-round argument of Appendix B.4 does go through round by round; what is missing is that the *input* to round $r + 1$ be in the restricted relation of Definition 3.25, which is a hypothesis on round r 's output that only the induction supplies. Second, the extraction side compounds independently, and badly, for the reason given in Remark 4.8.

Remark 4.7 (The folded structure must satisfy clause (4)). *Clause (4) of Definition 3.1, namely, that the circuit rows carry no support on the blinding columns, is a hypothesis on the structure $\mathbf{s}$ handed to $\Pi_{\text{LBF-Fold}}$, and it is worth checking rather than assuming. Whenever the layout is reached by appending $n_{\mathbb{F}, \text{bl}}$ fresh variables together with their $n_{\mathbb{F}, \text{bl}}$ fresh rows to an existing $\mathbf{s}^\circ$ it holds automatically, the pre-existing rows mentioning none of the new columns, and this is how every structure in this paper arises. The case to confirm is the recursive one. On iterating $\Pi_{\text{LBF-Fold}}$, the structure folded at round $r + 1$ encodes the verifier of round r , and that circuit is built first and padded afterwards, so clause (4) holds there by construction as well. It now does so for a stated reason rather than an implicit one, which is the point of recording it: an implementation that interleaved the blinding variables into the circuit's own variable numbering, instead of appending them, would meet clauses (1)–(3) and fail clause (4) silently.*

What would break is not a technicality. Without clause (4) the fibre of Lemma 3.29 depends on the circuit witness, so $\mathbf{u}_{\text{bl}}$ may fail on some inputs and truncation is no longer well defined. The transfer of knowledge soundness along $\mathbf{u}_{\text{bl}}$ goes with it, along the need the distribution of $\mathbf{e}$ to be fixed in advance by $\mathbf{b}$ and $n_{\mathbb{R}, \text{bl}}$ alone.

Remark 4.8 (Single invocation versus iteration). *Step 5 of Protocol 8 certifies the salts of its input commitment only in the relaxed sense of Definition 3.22. By Lemma 3.6(2) the extractor obtains $\bar{\mathbf{s}}_j$ with $\|\bar{\mathbf{c}}\bar{\mathbf{s}}_j\|_2 \leq \beta_{\text{tx}}$, not $\|\bar{\mathbf{s}}_j\|_\infty < B_{\text{fold}}$. For a single invocation of Π_{LBF} (the main use case this work is primarily aimed at, where folding is used once to introduce zero-knowledge on top of a given R1CS circuit) this is exactly what is needed, for three reasons. The output witness of Protocol 8 is the honest prover's, so its norms are a hypothesis of the reduction-of-knowledge definition (membership of (u_2, w_2) in $\mathcal{R}_2$) rather than a claim to be certified. The extracted R1CS witness z_1 , which is the object one actually cares about, is bounded exactly and within the same round by the norm-check polynomial $\text{NC}(\vec{X})$ of Protocol 6, Step 3, with no salt involved. And the remaining relaxed objects are the prover's own sampled blinding instances of Step 1 of Protocol 12. Indeed the final deliverable of Π_{LBF} is a witness for $\mathcal{R}' = \text{CCS}(\mathbf{b}, \mathcal{L})$ (Definition 2.21), which carries no salt component at all. More precisely, every salt clause in the intermediate relations is bookkeeping that keeps the composition type-correct, and its tightness never surfaces in the statement of Theorem 4.13. Relaxation therefore costs only the MSIS bound of Remark 3.23, and nothing at all in blinding.*

Iterating $\Pi_{\text{LBF-Fold}}$ is a different matter. There the witness at round r is not supplied by an honest prover but produced by the extractor of round $r + 1$, so its salt bound must be certified rather than assumed, and $\mathcal{R}_2$ -membership discharges the norm clauses only at the last round. The relaxation then compounds: each round contributes a factor $\bar{c}$ from Lemma 3.6(2) and a factor $(\rho_i - \rho'_i)$ from the coordinate-wise inversion of Lemma 2.19. The number of rounds supported is therefore $\mathcal{O}(1)$, and small.

To fold without such a bound, the fresh salts must be certified exactly. The device is the exact ℓ_2 -norm proof of [LNP22, §5.2], whose translation to $\mathbb{R}_{\mathbb{K}}$ is the content of §3.3.4, and which reaches the BDLOP randomness $\mathbf{s}_2$ (untouched by [LNP22]'s norm machinery, which targets the committed part $(\mathbf{s}_1, \mathbf{m})$) through the anchoring instantiation of §3.2.1.1. One would append it, at bound $\sqrt{kt}(\bar{B} - 1)\sqrt{(m_1 + m_2)d}$, to Step 5 above and, at bound $(\bar{B} - 1)\sqrt{t(m_1 + m_2)d}$, to the proof of knowledge at Step 17 of Protocol 7, whose mask salts are subject to the same objection. The extracted salts would then satisfy their bounds with no factor $\bar{c}$, binding of Commit would identify the certified opening at round r with the one the extractor of round $r + 1$ recovers, and nothing would accumulate. This is the salt-side analogue of the role $\text{NC}(\vec{X})$ already plays for the witness in [NS26], namely, an exact norm certificate, introduced precisely so that extraction need not divide by challenge differences.

4.3. Main results. We collect here all the results, and defer their proofs (for obtaining a public-coin, complete, sound, and zero-knowledge protocol) to Appendix B.

Lemma 4.9. *The R1CS reduction Π'_{R1CS} defined in Protocol 6 is a strong interactive reduction for the pair (ϕ, ψ) , where ϕ projects the commitments $(c_i)_{i \in [K+k]}$ from the instance and ψ projects the witness vectors*

$(\mathbf{z}_i)_{i \in [K+k]}$ from the witness, discarding the evaluation hints and the commitment salts, with completeness error $\varepsilon_c^{\text{R1CS}} \leq 5 \cdot 2^{-\lambda}$.

Proof. We provide the full proof in Appendix B.1. ■

Lemma 4.10. *The random linear combination reduction Π'_{RLC} defined in Protocol 7 is a weak interactive reduction for the same pair (ϕ, ψ) as in Lemma 4.9, and completeness error $\varepsilon_c^{\text{RLC}} \leq (1 + (K+k)t + t) \cdot 2^{-\lambda}$.*

Proof. We provide the full proof in Appendix B.2. ■

Lemma 4.11. *The composition $\Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$ is a reduction of knowledge, with completeness error $\varepsilon_c^{\text{R1CS}} + \varepsilon_c^{\text{RLC}} \leq (6 + (K+k+1)t) \cdot 2^{-\lambda}$.*

Proof. This is simply the combination of Lemma 4.9, Lemma 4.10, and Proposition 2.8. The composition clause of the latter is what lets us add the two completeness errors rather than re-deriving them: the composed honest execution runs Π'_{R1CS} and then Π'_{RLC} on its output, and fails only if one of the two does. ■

Lemma 4.12. *The decomposition reduction Π'_{DEC} defined in Protocol 8 is a reduction of knowledge from $\text{CE}_{\text{com}}(B, \mathcal{L}, B_{\text{fold}}, \text{Commit})(\text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit}))$ to $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$, with completeness error $\varepsilon_c^{\text{DEC}} \leq 2^{-\lambda}$ and negligible knowledge-soundness error ε_{DEC} (cf. Proposition 4.17), assuming the witness norm bound is satisfied (cf. Remark B.4.(1)).*

Proof. We provide the full proof in Appendix B.3. ■

Theorem 4.13. *The interactive reduction Π_{LBF} described in Protocol 12 is a reduction of knowledge, with completeness error $\varepsilon_c^{\text{LBF}} \leq (7 + (K+k+1)t)2^{-\lambda}$ (unbatched count, Proposition 4.15; $9 \cdot 2^{-\lambda}$ under the batched default of §4.1.2.1) and with knowledge-soundness error ε_{LBF} (Proposition 4.17).*

Assume moreover that $K = 1$, that the structure follows the blinded layout of Definition 3.1, and that we restrict to $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k \subset \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$. Consider the precomposition by ι_{bl} in place, that is, (Definition 4.6) Π_{LBF} taken over the unpadded input relation $\mathcal{R}' = \text{CCS}^\circ(b)^K$ of Definition 3.28, with the padding applied by the prover as in Equation (3.19). Then there is an EPT simulator $\mathcal{S}'$ which, on input $(\text{pp}, \mathbf{s}, x_1)$ alone (neither the witness $\mathbf{z}_1$, nor the blinding block, nor the commitment c_1) satisfies, for every honest-but-curious public-coin $\mathcal{V}^*$ and every pair $(x_1, \mathbf{w}_{\text{circ}})$ output by a PPT $\mathcal{A}$,

$$\{(u_1, \text{st}, \mathbf{w})\} \cong \{(u'_1, \text{st}', \mathbf{w}') \leftarrow \mathcal{S}'(\text{pp}, \mathbf{s}, x_1)\},$$

where $\mathbf{e} \leftarrow \chi_b^{n_{\text{R}, \text{bl}}}$ is drawn by the prover inside ι_{bl} and not by $\mathcal{A}$, and $\mathbf{z}_1 := [x_1, \mathbf{w}_{\text{circ}}, \mathbf{e}]$, $c_1 := \mathcal{L}(\mathbf{z}_1)$ are formed from it as prover messages. The left-hand side is the distribution of a real execution of Π_{LBF} (after the composition with ι_{bl}) read over all executions, aborting ones included (Remark 2.12), with distinguishing advantage at most

$$(4.6) \quad \varepsilon_{\text{LBF-blind}} := N_{\text{PoK}}^{\text{zk}} \varepsilon_{\text{ABDLOP}}^{\text{zk}} + (W_{\text{max}} + 2)t \varepsilon_{\text{hide}} + W_{\text{max}} \varepsilon_{\text{hide},0} + \mathcal{O}(W_{\text{max}} 2^{-\lambda}) + \varepsilon_{\mathcal{L}\text{-hide}},$$

where $N_{\text{PoK}}^{\text{zk}} := 4 + 2t$, and where $\varepsilon_{\text{hide}}$ is the MLWE hiding advantage of an ABDLOP commitment at the ambient (ℓ, m_2) while $\varepsilon_{\text{hide},0}$ is that of the mask-block commitment $C_{\mathbf{y},0}$, which is carried by a tuple of $\tilde{N}$ blocks and so contributes $\varepsilon_{\text{hide},0} = \tilde{N} \varepsilon_{\text{hide}}$ (Remark 4.21). In particular Π_{LBF} is blinding in the sense of Definition 2.11, and in the default regime (1) of Lemma 3.3 the term $\varepsilon_{\mathcal{L}\text{-hide}} \leq 2^{-128}$ is statistical and dominated by the others at $\lambda = 120$.

Proof. We provide the full proof in Appendix B.4. ■

Remark 4.14 (Composition Structure). *The choice of maintaining the decomposition of the protocols as in [NS26] is motivated from a theoretical point of view for ease of proving that the final protocol satisfies all the required properties. However, one can notice that the ABDLOP ZKPoK is called twice on the same inputs, which, in the combined protocol, this is unnecessary. Deduplicating removes the $(K+k)t$ invocations of the latter, so that N_{PoK} drops from $6 + (K+k+1)t$ to $6 + t$ and (cf. Proposition 4.15) $\varepsilon_c^{\text{LBF}} \leq (7+t)2^{-\lambda}$ accordingly; the $(K+k)t \varepsilon_{\text{ABDLOP}}^{(1)}$ term likewise leaves Equation (4.13). Combined with the batching of Remark 4.4 this brings N_{PoK} to 7 (§4.1.2.1).*

4.3.1. The consolidated completeness error. Unlike the reductions of [NS26], which are perfectly complete, ours abort with small probability, and from two unrelated causes: the Rej_1 mask loop of Protocol 7, and the rejection loop internal to *every* ABDLOP proof of knowledge.

Proposition 4.15 (Consolidated completeness error). *Let $\varepsilon_c^{\text{R1CS}}, \varepsilon_c^{\text{RLC}}, \varepsilon_c^{\text{DEC}}$ denote the completeness errors of Lemmas 4.9, 4.10 and 4.12. Then*

$$(4.7) \quad \varepsilon_c^{\text{R1CS}} \leq 5 \cdot 2^{-\lambda},$$

$$(4.8) \quad \varepsilon_c^{\text{RLC}} \leq \underbrace{2^{-\lambda}}_{\text{Rej}_1 \text{ mask loop}} + \underbrace{((K+k)t+t)2^{-\lambda}}_{\text{ABDLOP retries}},$$

$$(4.9) \quad \varepsilon_c^{\text{DEC}} \leq 2^{-\lambda},$$

and consequently $\Pi_{\text{LBF}} = \Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$, (and likewise $\Pi_{\text{LBF-Fold}}$, per round), are complete with error

$$(4.10) \quad \varepsilon_c^{\text{LBF}} \leq (1 + N_{\text{PoK}}) \cdot 2^{-\lambda} = (7 + (K+k+1)t) \cdot 2^{-\lambda}.$$

Proof. Equations (4.7)–(4.9) are the completeness clauses of the three lemmas cited, each a union bound over the failure events restricted to the reduction in question, plus, for Π'_{RLC} , the mask loop. The mask-loop bound is Remark 4.3. Equation (4.10) then follows from the composition clause of Proposition 2.8, applied twice.

Step 1 of Protocol 6 is a single $\Pi_{\text{many}}^{(1)}$ call on the concatenation of the kt salt blocks and so carries one rejection loop, whence the 5 of Equation (4.7). Steps 1 and 17 of Protocol 7 are written as $(K+k)t$ and t separate calls and so carry that many, hence Equation (4.8). The asymmetry is one of exposition, not of substance; the batched figures and the convention under which all of these counts are reported are fixed in §4.1.2.1.

Two independence claims are used and are worth isolating, since they are what makes the union bound legitimate rather than merely convenient. First, distinct invocations use independently sampled masks $(\mathbf{y}_1, \mathbf{y}_2)$, so their abort events are independent. This holds even for invocations that share a commitment, since we freeze the commitment and resamples only the mask. Second, the ABDLOP retry budgets are nested strictly inside the mask loop of Protocol 7 and cannot interact with it. A proof-of-knowledge retry never resamples the folding challenges $\vec{\rho}$, and a mask-loop iteration never resamples a proof-of-knowledge challenge c . ■

4.3.2. The consolidated knowledge-soundness error. The knowledge-soundness errors of $\Pi'_{\text{R1CS}}, \Pi'_{\text{RLC}}$ and Π'_{DEC} are described source by source, inside the respective proofs. We collect them here in one place, both because the composition of Proposition 2.8 adds them and because the resulting expression is what would lead to a suitable parameter choice. We stress that the terms below are *knowledge-soundness* terms only: the MLWE hiding advantages are blinding terms and are accounted for separately, in Theorem 4.13.

We first address the tensor batching procedure.

Lemma 4.16 (Schwartz–Zippel for the tensored batching). *Fix a structure $\mathbf{s}$, an input tuple to Protocol 6, and regard*

$$P := \mathbf{T} - \sum_{\vec{x} \in \{0,1\}^{\log_2 m}} Q(\vec{x})$$

as a formal polynomial in the challenge variables $(\vec{\gamma}^{(1)}, \vec{\gamma}^{(2)}, \vec{\gamma}^{(3)}, \delta_0, \delta_1, \vec{\alpha})$, the prover’s data $\{z_i\}, \{y_{i,j}\}$ and the point r being held fixed. Then

- (1) $\deg P \leq \max\{\log_2(m) + 2, 4\} = \log_2(m) + 2$ for $\log_2(m) \geq 2$; and
- (2) $P \equiv 0$ if and only if all three families of claims hold simultaneously, namely

$$\text{cf}(y_{i,j})_\ell = \text{cf}(\widetilde{M_j z_i})_\ell(r) \quad \forall (i, j, \ell), \quad f(\widetilde{M_1 z_i}, \dots, \widetilde{M_t z_i}) \in \mathcal{ZS}_{\log_2 m} \quad \forall i \in [K],$$

$$\prod_{j=-b}^{b-1} (\widetilde{z_i} - j) \in \mathcal{ZS}_{\log_2 m} \quad \forall i \in [K+k].$$

Consequently, if any one of those claims fails, then over $(\vec{\gamma}^{(1)}, \vec{\gamma}^{(2)}, \vec{\gamma}^{(3)}, \delta_0, \delta_1, \vec{\alpha})$ drawn uniformly and independently from $\mathbb{K}$,

$$\varepsilon_{\text{SZ}} := \Pr[P = 0] \leq \frac{\log_2(m) + 2}{|\mathbb{K}|}.$$

Proof. Using $\sum_{\vec{x} \in \{0,1\}^{\log_2 m}} \text{eq}(\vec{x}, \vec{\beta}) g(\vec{x}) = \tilde{g}(\vec{\beta})$ for both $\vec{\beta} = \vec{\alpha}$ and $\vec{\beta} = r$, and writing $f_i := f(\widetilde{M_1 z_i}, \dots, \widetilde{M_t z_i})$, $\text{NC}_i := \prod_j (\tilde{z}_i - j)$ and $E_{i,j,\ell} := \text{cf}(y_{i,j})_\ell - \text{cf}(\widetilde{M_j z_i})_\ell(r)$, one gets

$$(4.11) \quad P = \delta_1 \sum_{i=K+1}^{K+k} \sum_{j=1}^t \sum_{\ell=1}^d \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)} E_{i,j,\ell} - \sum_{i=1}^K \gamma_i^{(1)} \tilde{f}_i(\vec{\alpha}) - \delta_0 \sum_{i=1}^{K+k} \gamma_i^{(1)} \widetilde{\text{NC}}_i(\vec{\alpha}).$$

(1) The three groups of monomials have total degrees 4, $1 + \log_2(m)$ and $2 + \log_2(m)$ respectively, the $\vec{\alpha}$ -dependence being multilinear in each case and the Eval group carrying no $\vec{\alpha}$ at all (its eq factor is taken at the public point r , not at $\vec{\alpha}$).

(2) The monomials $\delta_1 \gamma_i^{(1)} \gamma_j^{(2)} \gamma_\ell^{(3)}$, $\gamma_i^{(1)}$ and $\delta_0 \gamma_i^{(1)}$ occurring in (4.11) are pairwise distinct: the three families are separated by the $\{\delta_1\}$, $\{1\}$, $\{\delta_0\}$ factors, and within each family the index triple (resp. index) is recovered from the monomial. Hence $P \equiv 0$ forces every coefficient to vanish, i.e. $E_{i,j,\ell} = 0$ for all (i, j, ℓ) , $\tilde{f}_i \equiv 0$ and $\widetilde{\text{NC}}_i \equiv 0$ as polynomials in $\vec{\alpha}$; and a multilinear extension vanishes identically iff the underlying function vanishes on $\{0, 1\}^{\log_2 m}$, which is membership in $\mathcal{ZS}_{\log_2 m}$. The converse is immediate from (4.11). The final bound is Schwartz–Zippel applied to the nonzero P over the uniform product distribution on $\mathbb{K}$. ■

Note also that item (2) needs no injectivity hypothesis on an index map, and in particular no analogue of a value separating F and NC from Eval: the δ -separation of Remark 4.2.(4) does that work.

We write $\ell := \log_2(m)$ for the number of Sum-Check variables and $D_{\max}$ as in §2.1. Then we have (cf. [NS26, Appendix B]):

- $\varepsilon_{\text{SZ}} := (\log_2(m)+2)/|\mathbb{K}|$: the Schwartz–Zippel error of the tensored batching challenges $\vec{\gamma}^{(1)}, \vec{\gamma}^{(2)}, \vec{\gamma}^{(3)}, \delta_0, \delta_1, \vec{\alpha}$ of Protocol 6, Step 2, i.e. the probability that the nonzero polynomial P of Lemma 4.16 vanishes at the random evaluation point;
- $\varepsilon_{\text{SC}} := \ell \cdot D_{\max}/|\mathbb{K}|$: the Sum-Check protocol of Protocol 6, Step 8;
- $\varepsilon_\zeta := 1/|\mathbb{K}|$: the blinding challenge ζ of Protocol 6 at Step 5, anchoring $\zeta P + T$;
- $\varepsilon_{\text{deg } 0} := 2/|\mathbb{K}|$: the batched degree-0 check (Lemma 3.13) of Protocol 6;
- $\varepsilon_{\text{ct}} := 1/|\mathbb{K}|$: one invocation of $\Pi_{\text{many}}^{(\text{ct})}$ (Lemma 3.8); there are three in Protocol 6, at Steps 7, 12 and 18;
- $\varepsilon_{\text{ABDLOP}}^{(1)}, \varepsilon_{\text{ABDLOP}}^{(2)}$: the knowledge-soundness errors of the linear and quadratic ABDLOP proofs of knowledge (Lemmas 3.6, 3.7).
- $\epsilon(B, \mathcal{C})$: the $(2B, \mathcal{C})$ -relaxed-binding error of the Ajtai commitment $\mathcal{L}$;
- $\epsilon(\text{Commit}, \beta_{\text{rlx}}, \mathcal{C})$: the binding error of Commit at the relaxed bound β_{rlx} of Definition 3.22, and $\epsilon(\text{Commit}, \tilde{B}, \mathcal{C})$ at the fresh-salt bound;
- $\epsilon_{\text{bind}}(\tilde{B})$: the $\tilde{B}$ -binding error of Commit from the two-opening argument for $C_{\mathbf{y},0}$ (Remark B.9). It is read at the ambient $\text{MSIS}_{\kappa, m_1+m_2, 2\tilde{B}\sqrt{(m_1+m_2)d}}$, and incurred once per block of the mask block, for $\tilde{N}$ in all; cf. Remark 4.21;
- $\varepsilon_{\text{cw}} := W_{\max} \cdot (K+k)/|\mathcal{C}|$: the coordinate-wise extraction loss of Lemma 2.19 with $\ell = K+k$, inflated by the attempt budget $W_{\max}$ of Protocol 7, which bounds the number of challenge blocks a malicious prover may select among before completing an attempt.

Proposition 4.17 (Consolidated knowledge-soundness error). *Let $\varepsilon_{\text{R1CS}}, \varepsilon_{\text{RLC}}, \varepsilon_{\text{DEC}}$ denote the knowledge-soundness errors of Lemmas 4.9, 4.10 and 4.12. Then*

$$(4.12) \quad \begin{aligned} \varepsilon_{\text{R1CS}} &\leq \underbrace{\varepsilon_{\text{SZ}} + \varepsilon_{\text{SC}}}_{\text{cf. [NS26, Appendix B.2]}} + \underbrace{\varepsilon_\zeta + \varepsilon_{\text{deg } 0} + 3\varepsilon_{\text{ct}} + 4\varepsilon_{\text{ABDLOP}}^{(1)} + \varepsilon_{\text{ABDLOP}}^{(2)}}_{\text{our modifications}} \\ &= \frac{\log_2(m)(D_{\max} + 1) + 8}{|\mathbb{K}|} + 4\varepsilon_{\text{ABDLOP}}^{(1)} + \varepsilon_{\text{ABDLOP}}^{(2)}, \end{aligned}$$

$$(4.13) \quad \varepsilon_{\text{RLC}} \leq \varepsilon_{\text{cw}} + \epsilon(B, \mathcal{C}) + \epsilon(\text{Commit}, \beta_{\text{rlx}}, \mathcal{C}) + \epsilon_{\text{bind}}(\tilde{B}) + (K+k+1)t\varepsilon_{\text{ABDLOP}}^{(1)},$$

$$(4.14) \quad \varepsilon_{\text{DEC}} \leq \varepsilon_{\text{ABDLOP}}^{(1)} + \epsilon(\text{Commit}, \tilde{B}, \mathcal{C}).$$

Consequently, the composed reduction $\Pi_{\text{LBF}} = \Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$ has knowledge-soundness error

$$(4.15) \quad \varepsilon_{\text{LBF}} \leq \varepsilon_{\text{R1CS}} + \varepsilon_{\text{RLC}} + \varepsilon_{\text{DEC}}$$

$$\begin{aligned}
&= \frac{\log_2(m)(D_{\max} + 1) + 8}{|\mathbb{K}|} + \frac{W_{\max}(K + k)}{|\mathcal{C}|} \\
&+ N_{\text{PoK}}^{(1)} \varepsilon_{\text{ABDLOP}}^{(1)} + \varepsilon_{\text{ABDLOP}}^{(2)} + \epsilon(B, \mathcal{C}) + \epsilon(\text{Commit}, \beta_{\text{rlx}}, \mathcal{C}) + \epsilon(\text{Commit}, \tilde{B}, \mathcal{C}) + \epsilon_{\text{bind}}(\tilde{B}),
\end{aligned}$$

where $N_{\text{PoK}}^{(1)} = 5 + (K + k + 1)t$, and which is negligible in λ whenever MSIS is hard for the relevant bounds and $|\mathbb{K}|, |\mathcal{C}|$ are superpolynomial.

Proof. Each summand is the error of the corresponding step, as established in the proofs of the cited lemmas, with the respective multiplicities. Within Protocol 6 the four $\varepsilon_{\text{ABDLOP}}^{(1)}$ arise from Step 1, from the two anchoring calls of Steps 7 and 18 (Lemma 3.16), and from Step 12; the last three are the inner $\Pi_{\text{many}}^{(1)}$ calls that the $\Pi_{\text{many}}^{(\text{ct})}$ wrappers make at their own Step 5, and they are also what the three ε_{ct} accompany. The remaining $(K + k + 1)t$ come from Steps 1 and 17 of Protocol 7, through which the extractor of Lemma 4.10 recovers the salts, and one from Step 5 of Protocol 8, giving $N_{\text{PoK}}^{(1)}$ in total. A union bound over the events gives Equations (4.12)–(4.13). The composition bound is [NS26, Theorem 12]. ■

We note one deliberate over-count. $\varepsilon_{\text{deg } 0} = 2/|\mathbb{K}|$ (Lemma 3.13) already absorbs the $\vec{\gamma}$ -aggregation loss of Step 12, and the $3\varepsilon_{\text{ct}}$ term counts that step a second time. We keep both, the bound being conservative and the difference $1/|\mathbb{K}|$ immaterial against ε_{SZ} .

4.3.3. Efficiency measures. We provide here a brief account of the complexity of the LatticeBlindFold protocol and we compare it to SuperNeo and Nova, the most closely related folding schemes. Considering only the complexity in terms of constraint sizes, that is, number of constraints $m = n_{\mathbb{F}}$, decomposition number $k = \Theta(\log_2 m)$, and sizes $n_{\mathbb{F}}$ and $n_{\mathbb{F}, \text{in}}$, we obtain the following Table 1.

Let us recall that the value $k \approx \log_2(m)$ used for both SuperNeo and LatticeBlindFold below is not a free choice. It is the decomposition number that the rejection sampling underlying Π_{LBF} 's blinding forces upon it (Remark 4.18). We report Π_{SuperNeo} at this same k purely so that the two columns describe the same protocol structure and can be compared metric by metric. However, Π_{SuperNeo} carries no such obligation. It only needs $k = \Theta(1)$, at which its Prover Time, Verifier Time, and Communication all improve by a further $\Theta(\log_2(m))$ factor over what the table shows. That gap is, in essence, the structural cost of adding blinding, on top of the additive $n_{\mathbb{F}, \text{in}}$ term discussed below. A second, less visible cost sits inside the constants rather than in the table: Π_{LBF} transmits the public part of the k blinding pairs its prover samples (k compact Ajtai commitments and kt evaluation hints), which Π_{SuperNeo} has no analogue of. Being $\Theta(k) = \Theta(\log_2(m))$, this is absorbed by the $\log_2(m)$ term already present in the Communication row and by the $k\kappa d$ contribution, so the asymptotics below stand as written. The blinding block of Definition 3.1 enters every count only through $n_{\mathbb{F}}$, which it increases by $n_{\mathbb{F}, \text{bl}} = \Theta(\kappa d \log q)$.

Metric	LatticeBlindFold (Small Fields)	SuperNeo (Small Fields)	Nova (Large Field)
Prover Time	$\mathcal{O}(n_{\mathbb{F}} \log_2(m))$	$\mathcal{O}(n_{\mathbb{F}} \log_2(m))$	$\mathcal{O}(m)$
Verifier Time	$\mathcal{O}(n_{\mathbb{F}, \text{in}} \log_2(m))$	$\mathcal{O}(n_{\mathbb{F}, \text{in}} \log_2(m))$	$\mathcal{O}(n_{\mathbb{F}, \text{in}})$
Communication	$\mathcal{O}(\log_2(m) + n_{\mathbb{F}, \text{in}})$	$\mathcal{O}(\log_2(m))$	$\mathcal{O}(1)$

TABLE 1. Asymptotic complexities with respect to circuit sizes m , $n_{\mathbb{F}}$, and $n_{\mathbb{F}, \text{in}}$ only, treating d and q (with $\lambda_{\text{Nova}} \approx 4 \times \log_2(q)$) as constants. Both lattice schemes are shown at the matched decomposition number $k \approx \log_2(m)$ that Π_{LBF} 's rejection sampling requires; Π_{SuperNeo} needs only $k = \Theta(1)$ and is correspondingly $\Theta(\log_2(m))$ cheaper on its own terms (Remark 4.18).

Table 1 is asymptotic, and each column is counted in its own primitive. A translation of $\mathbb{G}$ -operations for Nova, $\mathbb{R}_{\mathbb{F}}$ - and $\mathbb{K}$ -multiplications for SuperNeo and for Π_{LBF} . They are therefore not directly comparable as constants: a 256-bit base-field multiplication is some 16 word multiplications against one for $\mathbb{F}$ at $q < 2^{64}$, and an $\mathbb{R}_{\mathbb{F}}$ -multiplication is $\Theta(d \log d)$ of the latter. The size parameters differ too as Nova is linear in the

number of nonzero entries of the constraint matrices (here approximated to $\Omega(m)$), whereas the Sum-Check of Π_{LBF} is linear in the padded square dimension $m = n_{\mathbb{F}}$, which absorbs both the power-of-two padding and the blinding block of Definition 3.1. The two agree up to a constant for circuit-derived R1CS.

Remark 4.18 (The growth of k , and the basis of the comparison). *The number k of decomposition pieces is fixed by the norm bound $B = b^k$ that the folded witness must respect, not by the folding depth. For Π_{SuperNeo} the requirement is*

$$(4.16) \quad (K + k) T(\mathcal{C}) (b - 1) < b^k$$

(Definition 2.2), whereas for Π_{LBF} the rejection sampling of Π'_{RLC} inflates it to

$$(4.17) \quad \tau_{\lambda, \xi} \xi (K + k) T(\mathcal{C}) (b - 1) \sqrt{n_{\mathbb{F}}} < b^k$$

(Lemma B.1, §3.6). The factor $\sqrt{n_{\mathbb{F}}}$ in (4.17) forces

$$k \approx \frac{1}{2} \log_2(n_{\mathbb{F}}) + \log_2(\tau_{\lambda, \xi} \xi (K + k) T(\mathcal{C}) (b - 1)) = \Theta(\log n_{\mathbb{F}}).$$

We obtain the values $k = 29$ (at $n_{\mathbb{F}} = 2^{20}$) and $k = 30, 31$ (at the $n_{\mathbb{F}} = 2^{21}$) tabulated in §3.6. On the other hand, Equation (4.16) is met by a k which is independent of $n_{\mathbb{F}}$; at $T(\mathcal{C}) = 128$, $b = 2$, $K = 1$ one may take $k = 13$, and $k = \Theta(1)$ asymptotically.

Table 1 reports both lattice schemes with $k = \Theta(\log n_{\mathbb{F}})$ that Equation (4.17) imposes on us. Every $\log_2(n_{\mathbb{F}})$ factor appearing there is a k -factor, and it is charged to both schemes alike. Since Π_{SuperNeo} is not obliged to run at this k , but at the smaller $k = \Theta(1)$ permitted by Equation (4.16) its costs improve to

$$\text{Prover } \mathcal{O}(d n_{\mathbb{F}}), \quad \text{Verifier } \mathcal{O}(d + n_{\mathbb{F}, \text{in}} + \log_2(m)), \quad \text{Communication } \mathcal{O}(d + \log_2(m)),$$

i.e. a $\Theta(\log n_{\mathbb{F}})$ improvement across the board. That gap is the price of blinding, and we wish to stress it here.

Unfortunately, the constants in the asymptotic complexity are bigger for the LatticeBlindFold protocol, in particular, the proof size grows due to the masking techniques we employ (cf. §4.3.3.1). However, if we are interested in introducing zero-knowledge on top of a protocol, we can wrap the initial protocol in an efficient SNARK and only then apply our LatticeBlindFold technique, so that the $n_{\mathbb{F}}$ these figures scale in is the SNARK verifier's circuit rather than the original computation's. The sizes above are the cost of producing the folded claims, not of checking them, and the two should not be confused. Π_{LBF} terminates with k pairs in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)$ and with the transcripts of $N_{\text{PoK}}^{(1)} = 5 + (K + k + 1)t$ ABDLOP proofs of knowledge; a verifier convinced by them is convinced of the original R1CS statement, but only after checking them. Read as NovaBlindFold is read, this leaves a decider to be supplied, and there are two ways to supply it. The direct one hands the openings to the downstream verifier unchanged and a linear scan of them. The succinct one arithmetizes the ABDLOP verification equations $\mathbf{t}_A = \mathbf{A}_1 \mathbf{s}_1 + \mathbf{A}_2 \mathbf{s}_2$, $\mathbf{t}_B = \mathbf{B} \mathbf{s}_2 + \mathbf{m}$ together with the masked-opening norm checks, and proves them inside a further R1CS instance; that circuit is $\Theta(N_c(m_1 + m_2)d)$ ring operations and the norm checks are its awkward part, being the same exact- ℓ_2 machinery Remark 4.8 needs. We instantiate neither, and record the gap as such.

4.3.3.1. A concrete size estimate. At the two parameter sets of §4.3.4 ($q = 2^{64} - 59$, $\mathbb{K} = \mathbb{F}_{q^2}$, $K = b - 1 = 1$, $t = 3$, $n_{\mathbb{F}} = 2^{21}$, $\xi \approx 20$, with $k = 31$ at $d = 128$ and $k = 30$ at $d = 64$), a single folding step costs ≈ 8.2 resp. ≈ 5.5 MB, distributed as follows.

	$d = 128$	$d = 64$
masked openings	4.57 MB	3.01 MB
blocks $\sum N_c$	246	238
ABDLOP commitments, output	1.98 MB	1.19 MB
ABDLOP commitments, sampled input	0.70 MB	0.42 MB
compact Ajtai commitments, output	0.47 MB	0.39 MB
compact Ajtai commitments, sampled input	0.45 MB	0.38 MB
first messages w , challenges, garbage terms	0.07 MB	0.06 MB
total	≈ 8.2 MB	≈ 5.5 MB

The commitment counts behind the two columns are $88 + 31$ ABDLOP and $32 + 31$ compact Ajtai at $d = 128$, against $85 + 30$ and $31 + 30$ at $d = 64$. The unit costs are 23.05 KB and 14.38 KB per ABDLOP commitment

respectively, and 15 KB resp. 13 KB per compact Ajtai commitment; the ABDLOP figure is $\kappa d(\log_2 q - D)$ bits for the compressed $\mathbf{t}_A$ together with $\ell d \log_2 q$ for $\mathbf{t}_B$, the compact Ajtai one $\kappa_{\text{Ajtai-Compact}} d \log_2 q$. The gap between the columns is not the packing width, which is identical at the two degrees: both sets run at $\ell = 16$, hence at $\ell/2 = 8$ $\mathbb{R}_{\mathbb{K}}$ -messages per commitment, and the counts above differ only through k . Nor does the halved ring degree cancel, notwithstanding that a full-width ring element is 0.5 KB at $d = 64$ against 1 KB at $d = 128$. It cancels only where the dimension is MSIS/MLWE-driven and so roughly doubles as d halves: the compact Ajtai commitment moves by 13% ($\kappa_{\text{Ajtai-Compact}}$ going $15 \rightarrow 26$) and the $\mathbf{t}_A$ part of an ABDLOP commitment by 9% (κ going $11 \rightarrow 19$). Wherever the dimension is instead fixed independently of d it does not cancel at all: $\mathbf{t}_B$ costs $\ell d \log_2 q$ at $\ell = 16$ and so halves outright, from 131 072 to 65 536 bits, and the $\mathbf{z}_1$ half of every masked opening, at $m_1 = 15$, halves with it. Since the object counts are set by the instance index ($K + k$ and k) rather than by ring capacity, they barely move (119 ABDLOP commitments against 115) and the per-object saving is not bought back. Against that saving, the $d = 64$ set delivers ≈ 116 bits rather than ≈ 121 (Remark 4.19), so the two rows are not interchangeable. The two commitment rows count both the objects the reduction outputs and the public part of the k blinding pairs the prover samples at Step 1 of Protocol 12, which $\mathcal{V}$ must receive in order to fold; the latter account for ≈ 1.15 MB of the total. Their public inputs $\mathbf{x}_i^{\text{blind}}$ are not charged separately, as $\mathcal{V}$ derives all k of them from one transmitted instance. One call in particular sets the widths. Of the eight batched ABDLOP invocations a folding step makes, exactly one masks a *folded* salt: Step 5 of Protocol 8, whose input C_j is the output of Π'_{RLC} and carries $\|\mathbf{s}_l\|_{\infty} \leq B_{\text{fold}} - 1$ by Lemma B.1. Every other call (the sampler of Protocol 11, Steps 9–10 of Protocol 6, the mask salts, and the output of a previous Π'_{DEC} , which resamples) sees ternary salts only, and this is exactly what the resampling of Remark 4.5 buys. The widths must therefore be read per call: the folded call runs at $\log_2 \mathfrak{s}_1 = 28.58$ against ≈ 18.1 for the fresh ones at $d = 128$. It is the widest of the eight, but neither the longest ($N_c = 32$, against the 53 of the Step 18 anchoring call) nor the largest single contributor, accounting for 861 KB of the 4675 KB of masked openings against the 955 KB of Step 18. What singles it out is the cost per block, 26.9 KB against 18.0 KB, and that is the price of the folded salt.

We conclude by remarking that we achieve zero-knowledge and plausible post-quantum hardness at the expense of bigger proofs and an overburdened prover (when compared to SuperNeo and Nova), but that such trade-off is less steep when we keep in mind that Nova needs to work over a large 254-bit field, while our LatticeBlindFold protocol works over small fields of ≈ 64 bits. Moreover, the lattice-framework is prone to improvements based on parallelization, which lends itself to GPU-based speedups; possible improvements are likely to be found in techniques devised by the [Midnight Foundation](#) in their implementation of Nightstream [Fou25] (and SuperNeo [NS26]).

4.3.4. Choice of parameters. We record here two explicit instantiations, kept as close as possible to those of [NS26, §8]. The Ajtai-side MSIS security is estimated with the SageMath [The25] routines of [NS26, Appendix B.5–B.6] (cf. [APS15]), and the combined MLWE/MSIS security on the ABDLOP side with the code of [LNP22, §6.1] (<https://github.com/khanhcrypto/LBZKP>), from which we drop the approximate range proofs ([LNP22, Lemmas 2.7, 2.9]) and approximate

$$\varepsilon_{\text{ABDLOP}}^{(1)} \approx \varepsilon_{\text{ABDLOP}}^{(2)} \leq \frac{2}{|\mathcal{C}_{\text{ABDLOP}}|} + q^{-d/2} + q^{-\lambda_{\text{ABDLOP}}},$$

with $\mathcal{C}_{\text{ABDLOP}}$ and λ_{ABDLOP} the strong sampling set and security parameter of the ABDLOP scheme. Throughout we take the Solinas prime $q = 2^{64} - 59$, a padded circuit of $m = n_{\mathbb{F}} = 2^{21}$, $\xi \approx 20$, $K = (b - 1) = (\tilde{B} - 1) = 1$, and $\mathcal{C}$ the set of polynomials with coefficients in $\{-1, 0, 1, 2\}$, so that $|\mathcal{C}| = 4^d$. The decomposition number k is then forced by §3.6, and T , for the ABDLOP commitments, is the negacyclic bound of Remark 2.17 rather than the weaker Proposition 2.16.

One reading convention applies to both sets. A folding step makes eight batched ABDLOP invocations on concatenated instances (Definition 3.15), whose masking widths are not a single pair: they depend on the concatenation length N_c and on the number n_{fold} of blocks carrying a *folded* salt, through

$$(4.18) \quad \alpha^{(l)}(N_c, n_{\text{fold}}) = \sqrt{n_{\text{fold}}(B_{\text{fold}} - 1)^2 + (N_c - n_{\text{fold}})(\tilde{B} - 1)^2} \sqrt{m_l d}, \quad l \in [2],$$

with $\mathfrak{s}_1 = \gamma_1 \eta \alpha^{(1)}$ and $\mathfrak{s}_2 = \gamma_2 \eta \alpha^{(2)}$. Exactly one call has $n_{\text{fold}} > 0$, namely Step 5 of Protocol 8, whose input commitment is the output of Π'_{RLC} and so carries $\|\mathbf{s}_l\|_{\infty} \leq B_{\text{fold}} - 1$ by Lemma B.1; every other call sees ternary salts only. That the salt budget is decoupled from k in this way is exactly what the resampling of Remark 4.5 buys, and it is why the figures below are a per-call sweep rather than one pair of standard

deviations. Accordingly, every MSIS figure we quote is the *maximum over all eight invocations*, the folded call included; a single-call estimate does not certify that call.

The sweep covers those eight invocations and nothing else. In particular it does *not* cover the mask-block commitment $C_{\mathbf{y},0}$ of Protocol 7, which is a second ABDLOP instance at a different message length and a different salt width, and which is moreover the one commitment of the protocol whose salts are opened in the clear. Remark 4.21 brings it back inside the sweep, by splitting its message across a tuple of commitments at the ambient parameters rather than widening a single one; what the circuit then controls is the number $\tilde{N}$ of blocks, not any lattice dimension.

	$d = 64$	$d = 128$
$\eta, \Phi_\eta(x)$	$128, x^{64} + 1$	$256, x^{128} + 1$
$ \mathcal{C} , T$	$2^{128}, 128$	$2^{256}, 256$
$k, B_{\text{fold}} = (K + k)T(\tilde{B} - 1) + \tilde{B}$	30, 3970	31, 8194
$\kappa_{\text{Ajtai-Compact}}$	26	15
$\kappa, \kappa' = \ell, \kappa_{\text{com}}$	19, 16, 35	11, 16, 27
m_1, m_2 ($m_1 d \geq 640$)	15, 83	15, 53
$\tilde{N}$, mask-block blocks (Rmk. 4.21)	$\lceil (19 + n_{\text{R},\text{in}})/16 \rceil$	$\lceil (11 + n_{\text{R},\text{in}})/16 \rceil$
$m_1 + m_2$; ceiling for statistical binding	98; 261	68; 151
$n_{\text{R},\text{bl}}, n_{\mathbb{F},\text{bl}}$ (regime (1))	1668, 106 752	962, 123 136
$\lambda_{\text{ABDLOP}}, \beta_{\text{ch}}$	150, 9	256, 2
$\gamma, D, M_{\text{PoK}}$	$2^{30}, 21, 5.37$	$2^{35}, 23, 5.37$
MSIS/MLWE dimensions	19, 40	11, 20
$\delta_{\text{MSIS}}, \delta_{\text{MLWE}}$	1.004419, 1.004399	1.004491, 1.004399
$\log_2 \mathfrak{s}_1$ (fresh; folded)	15.08–17.92; 27.04	15.58–18.45; 28.58
$\log_2 \mathfrak{s}_2$ (fresh; folded)	13.07; 25.02	13.25; 26.25
Ajtai MSIS	$2^{-144.5}$	$2^{-148.6}$
$\varepsilon_{\text{ABDLOP}}^{(1)}, N_{\text{PoK}}^{(1)} = 5 + (K + k + 1)t$	$2^{-134.9}, 101$	$2^{-147.6}, 104$
$N_{\text{PoK}}^{(1)} \varepsilon_{\text{ABDLOP}}^{(1)}$	$2^{-128.2}$	$2^{-140.9}$
$\varepsilon_{\text{SZ}}, \varepsilon_{\text{SC}}, \varepsilon_{\text{deg } 0}$	$2^{-123.5}, 2^{-121.6}, 2^{-127}$	$2^{-123.5}, 2^{-121.6}, 2^{-127}$
$W_{\text{max}}(K + k)/ \mathcal{C} $	$2^{-116.1}$	$2^{-244.1}$
binding term	$W_{\text{max}}(K + k)/ \mathcal{C} $	ε_{SC}
overall (interactive)	≈ 116 bits	≈ 121 bits

TABLE 2. The two illustrative parameter sets, at $q = 2^{64} - 59$, $n_{\mathbb{F}} = 2^{21}$, $\xi \approx 20$, $t = 3$, $K = b - 1 = \tilde{B} - 1 = 1$, $\lambda = 120$. All errors are the consolidated knowledge-soundness terms of Proposition 4.17, and all figures are *interactive* (§4.5). The two mask-block rows are the exception to everything else here: they are not outputs of the eight-call sweep but the separately fixed parameters of $C_{\mathbf{y},0}$, and they depend on the circuit through $n_{\text{R},\text{in}}$ (Remark 4.21).

In neither case is the lattice layer what binds. Both sets meet the $\delta < 1.0045$ targeted in [LNP22, §6.1], and at $d = 128$ the returned $\delta_{\text{MSIS}} = 1.004491$ clears it only narrowly: the search returns the smallest admissible dimension, and a practitioner should prefer $\kappa = 12$, at ≈ 0.6 KB per commitment, rather than sit at the boundary. Since $\ell = 16$ is even and $\mathbb{K}/\mathbb{F}$ is quadratic, each of these commitments carries exactly $\ell/2 = 8$ elements of $\text{R}_{\mathbb{K}}$ with no slot wasted. The parity remark applies to them alone: the blocks of $C_{\mathbf{y},0}$ carry native $\text{R}_{\mathbb{F}}$ -vectors, so no packing arises there and all ℓ slots of a block are used (Remark 4.21).

Remark 4.19 (Consistency at λ). *We fix $\lambda = 120$ as the working level, and every bit count above is an interactive figure.*

- (1) The $d = 64$ set reaches ≈ 116 bits and so falls short of $\lambda = 120$; it is illustrative and should not be taken as a secure instantiation. What limits it is the challenge-space term $W_{\max}(K + k)/|C|$ at $|C| \approx 2^{128}$, not the lattice layer, so raising it calls for a larger C rather than larger commitment dimensions, at the price of a larger $T(C)$ feeding back into B_{fold} and α . Comparing smaller base fields against higher-degree extensions would be interesting.
- (2) The $d = 128$ set reaches ≈ 121 bits under the independent-challenge default, the binding term being the Sum-Check error $\varepsilon_{\text{SC}} = \log_2(m)D_{\max}/|\mathbb{K}|$. Reaching $\lambda = 128$ would need $|\mathbb{K}| \gtrsim 2^{134}$, hence a cubic extension or a larger base prime.
- (3) The lattice figures follow the methodology of [LNP22, §6.1], which measures hardness by the root Hermite factor; cf. item (3) of §5.

Remark 4.20 (Cost of the blinding block). The block adds $n_{\mathbb{F},\text{bl}}$ columns and, the system being square, $n_{\mathbb{F},\text{bl}}$ rows. Two denominators are in play. Against an unpadded circuit of 2^{20} constraints, $n_{\mathbb{F},\text{bl}} = 123\,136$ at $d = 128$ is $\approx 11.7\%$ (and $106\,752$ at $d = 64$ is $\approx 10.2\%$); against the padded $n_{\mathbb{F}} = 2^{21}$ it is $\approx 5.9\%$ (resp. $\approx 5.1\%$). The percentages quoted elsewhere in the paper refer to the first.

The distinction matters because the block is affordable only when it fits inside padding the circuit already carries: a circuit of exactly 2^{20} constraints does not leave room, and appending the block there pushes m past the power of two and doubles it, a 100% overhead rather than an 11% one. The sets above, stated at $n_{\mathbb{F}} = 2^{21}$, therefore describe a circuit whose proper size lies between $925\,441$ and $2^{21} - n_{\mathbb{F},\text{bl}} = 1\,974\,016$ constraints at $d = 128$. The block should accordingly be reserved by the frontend at circuit-synthesis time, not appended afterwards. Two levers shrink it: taking $b = 4$ (item (5) of §5) divides the requirement by $\log_2 6$, giving $n_{\mathbb{F},\text{bl}} = 47\,744$, about 4.6% of 2^{20} , and halves k at the same time; failing that, regime (2) of Lemma 3.3 needs $n_{\mathbb{R},\text{bl}} \approx 45$ at $d = 128$, about 0.5%, at the price of a computational rather than statistical hiding term.

Remark 4.21 (The mask-block commitment). Everything above is a per-call sweep over the eight batched invocations of a folding step, all of which run at $\ell = 16$. The mask-block commitment $C_{\mathbf{y},0}$ of Protocol 7 is not among them, and we cannot take it monolithically. By Remark B.3 it carries the pair $(c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}) \in \mathbb{R}_{\mathbb{F}}^{\kappa} \times \mathbb{R}_{\mathbb{F}}^{n_{\mathbf{R},\text{in}}}$ in BDLOP slots, so a single commitment would need a message length $\kappa + n_{\mathbf{R},\text{in}}$, which equals 16 only by coincidence. At $d = 64$ we have $\kappa = 19 > 16$, so this exceeds ℓ already at $n_{\mathbf{R},\text{in}} = 0$, while at $d = 128$, $\kappa = 11$ leaves five free slots and it exceeds ℓ as soon as $n_{\mathbf{R},\text{in}} > 5$.

The obstruction is not the mismatch itself but its shape: $\kappa + n_{\mathbf{R},\text{in}}$ depends on the circuit. A monolithic mask block would therefore not have a parameter set at all, but a family of them, one per input length, together with the salt width $m_2 + \kappa + n_{\mathbf{R},\text{in}} - \ell$ that preserving the MLWE secret dimension forces alongside it; and no finite number of estimator runs covers a family. We avoid this by splitting the message rather than widening the commitment.

Fix the public partition of $[\kappa + n_{\mathbf{R},\text{in}}]$ into

$$\tilde{N} := \lceil (\kappa + n_{\mathbf{R},\text{in}}) / \ell \rceil$$

consecutive blocks of ℓ slots, the last zero-padded, and commit to the $\tilde{n}$ -th block as $C_{\mathbf{y},0}^{(\tilde{n})} := \text{Commit}(\mathbf{m}^{(\tilde{n})}, \mathbf{s}_{1,\mathbf{y},0}^{(\tilde{n})}, \mathbf{s}_{2,\mathbf{y},0}^{(\tilde{n})})$ under its own fresh salt pair, all $\tilde{N}$ pairs being disjoint from the salts of $\{C_{\mathbf{y},j}\}$. Step 6 of Protocol 7 sends the tuple $(C_{\mathbf{y},0}^{(\tilde{n})})_{\tilde{n} \in [\tilde{N}]}$, Step 12 opens all $\tilde{N}$ of them, and $\mathcal{V}$ runs its recomputation and its norm check blockwise before reassembling $(c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}})$ by concatenation. Nothing else changes. The partition is public and fixed before any challenge is drawn, so the binding argument of Remark B.9 applies blockwise and pins the concatenation; and no compatibility constraint is violated, since $C_{\mathbf{y},0}$ is never combined homomorphically with any other commitment ($\mathcal{V}$'s Step 21 check involves only $\{C_{\mathbf{y},j}\}_{j \in [t]}$, and $c_{\mathbf{y}}$ enters the folded c through its opened value).

Every block is then an ABDLOP commitment at the ambient (ℓ, m_1, m_2) under the ambient matrices $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B})$, with three consequences.

- (1) Hiding is $\text{MLWE}_{\kappa+\ell, m_2-\kappa-\ell}$ and binding is $\text{MSIS}_{\kappa, m_1+m_2, 2\tilde{B}\sqrt{(m_1+m_2)d}}$. Note also that the blocks carry native $\mathbb{R}_{\mathbb{F}}$ -vectors.
- (2) As recorded in Remark B.9, this is the one commitment whose salts are opened, and the count is therefore over differences on the revealed-salt set. At $\tilde{B} - 1 = 1$ two accepting openings of a block differ by a vector with coefficients in $\{-2, \dots, 2\}$, of which there are at most $5^{(m_1+m_2)d}$; and since $q = 2^{64} - 59$ has multiplicative order 32 modulo 128 and 64 modulo 256, the polynomial $X^d + 1$ splits into $l = 2$ irreducible factors of degree $d/2$ modulo q at both $d = 64$ and $d = 128$, so a fixed nonzero

such vector is annihilated by a uniform $[\mathbf{A}_1 \mid \mathbf{A}_2]$ with probability at most $q^{-\kappa d/l}$. A union bound gives statistical binding at $\lambda = 120$ as soon as

$$(4.19) \quad (m_1 + m_2) d \log_2 5 + \lambda \leq \kappa (d/l) \log_2 q,$$

i.e. $m_1 + m_2 \leq 261$ at $d = 64$ and ≤ 151 at $d = 128$, against the actual 98 resp. 68. A further union bound over the $\tilde{N}$ blocks costs $\tilde{N} \epsilon_{\text{bind}}(\tilde{B})$ on a quantity with thousands of bits of margin.

- (3) A hybrid over the $\tilde{N}$ blocks, whose salts are independent, gives $\epsilon_{\text{hide},0} = \tilde{N} \cdot \epsilon_{\text{hide}}$ in Equation (4.6). What Theorem 4.13 consumes on a rejected attempt is plain MLWE hiding of $(\mathbf{t}_A, \mathbf{t}_B)$ and nothing more, the proof-of-knowledge simulator not being in play, so no rejection-sampling transcript about it is ever produced. Thus $n_{\text{R},\text{in}}$ enters the error budget as a coefficient rather than the lattice parameters as a dimension, which is the whole point of the construction.

Another option is to adopt a mixed approach. Fix a block length ℓ^{blk} independently of the circuit, say $\ell^{\text{blk}} = 64$, run the protocol at $(\kappa, \ell^{\text{blk}}, m_1, m_2 + \ell^{\text{blk}} - \ell)$, and take $\tilde{N} = \lceil (\kappa + n_{\text{R},\text{in}}) / \ell^{\text{blk}} \rceil$. That is one extra run in total rather than one per circuit and it cuts $\tilde{N}$ by four.

Remark 4.22 (The bound at which binding of Commit is assumed). The sweep above calibrates the widths $\mathfrak{s}_1, \mathfrak{s}_2$, i.e. the bounds at which an opening is certified. Proposition 4.17 also carries $\epsilon(\text{Commit}, \beta_{\text{rlx}}, \mathcal{C})$, the binding error at the relaxed bound of Definition 3.22, a different and larger quantity, $\beta_{\text{rlx}} = 2\mathfrak{s}_2 \sqrt{2N_c m_2 d}$ by Equation (3.14), which is nowhere else quantified. The folded block dominates $\mathfrak{s}_2$ wherever it occurs, so the bound is set by Step 5 of Protocol 8, at $\log_2 \beta_{\text{rlx}} = 36.6$ at $d = 128$ and 35.2 at $d = 64$, and not by the Step 18 anchoring call, which reaches only 26.8 resp. 26.4 despite 1.6 times the concatenation length. Binding is thus assumed at $\text{MSIS}_{\kappa, m_1 + m_2, 2\beta_{\text{rlx}}}$ with $2\beta_{\text{rlx}} \approx 2^{37.6}$ resp. $2^{36.2}$, against $q \approx 2^{64}$.

This needs no separate estimator run: the sweep already evaluates the root Hermite factor at the extracted bound $4\eta \sqrt{B_1^2 + B_2^2}$ of Lemma 3.6, and $B_2 \geq 2\mathfrak{s}_2 \sqrt{2N_c m_2 d} = \beta_{\text{rlx}}$, whence that bound exceeds $2\beta_{\text{rlx}}$ by a factor $\geq 2\eta$; since δ is increasing in the bound, binding at β_{rlx} is certified a fortiori. This covers β_{rlx} but not $\beta_{\text{rlx}}^{(3)}$ of Equation (3.15), where the extra factor $2T_1(\mathcal{C}) = 2^{9.6}$ at $d = 128$ exceeds the $\approx 2^{7.9}$ of slack available, so the estimator should be re-run at $2\beta_{\text{rlx}}^{(3)} \approx 2^{48.2}$ resp. $2^{46.4}$; both remain far below q and we expect no material change in δ , but we do not claim it a fortiori. Finally, β_{rlx} is the only bound in the paper that grows with N_c , hence with k, t and ℓ ; on the call that binds it scales as $\sqrt{N_c}$ rather than N_c , so the doubling of N_c that a quartic extension brings (§4.5) costs about half a bit, and more only insofar as m_2 grows with it.

Remark 4.23 (What sets the blinding bound). Equation (4.6) is worth reading for which term dominates, since the answer is not the one the shape of the expression suggests: the bound is set from the start by the attempt cap, the statistical term being $\mathcal{O}(W_{\text{max}} 2^{-\lambda}) = \mathcal{O}(2^{8-120}) = \mathcal{O}(2^{-112})$, an 8-bit loss off the nominal $2^{-\lambda}$ that nothing else comes within four bits of. Ranking the remaining terms at $t = 3$ and $\epsilon_{\text{hide}} \approx \epsilon_{\text{ABDLOP}}^{\text{zk}} \approx 2^{-128}$, with $\epsilon_{\mathcal{L}\text{-hide}} \leq 2^{-128}$ by Lemma 3.3:

$$\underbrace{2^{-112}}_{\mathcal{O}(W_{\text{max}} 2^{-\lambda})} \gg \underbrace{2^{-118}}_{(W_{\text{max}}(t+1)+2t) \epsilon_{\text{hide}}} \gg \underbrace{2^{-124.7}}_{N_{\text{PoK}}^{\text{zk}} \epsilon_{\text{ABDLOP}}^{\text{zk}}} \geq \underbrace{2^{-128}}_{\epsilon_{\mathcal{L}\text{-hide}}}.$$

The coefficient on the hiding terms is $(W_{\text{max}} + 2)t + W_{\text{max}} = 1030 \approx 2^{10}$, entirely dominated by the W_{max} -weighted part, so the top two terms are both W_{max} -driven. That count reads the mask block as a single commitment; by Remark 4.21 it is a tuple of $\tilde{N}$ blocks, so the coefficient is properly $(W_{\text{max}} + 2)t + W_{\text{max}} \tilde{N}$, which at $\tilde{N} = 6$ puts the hiding term at $\approx 2^{-117.7}$. It is worth keeping the $W_{\text{max}} \epsilon_{\text{hide},0}$ term of Equation (4.6) visible nonetheless, since it is the only one that scales with the length of the public input. The proof-of-knowledge term is the smallest of the three carrying a coefficient, and for a reason worth stating: at $t = 3$ the simulated count is $N_{\text{PoK}}^{\text{zk}} = 4 + 2t = 10$, against the $N_{\text{PoK}}^{(1)} = 104$ invocations entering Proposition 4.17, the great majority of the proofs of knowledge being executed honestly by $\mathcal{S}$ on openings it already holds; the masked Sum-Check likewise contributes nothing, being simulated exactly (§4.1.1.1).

Two consequences. First, tightening the blinding guarantee means lowering the repetition rate M of Protocol 7 (equivalently raising ξ), not deduplicating proofs of knowledge as Remark 4.14 does for soundness. Second, at these parameters Π_{LBF} delivers blinding at $\epsilon_{\text{LBF-blind}} \approx 2^{-112}$ rather than at the nominal 2^{-120} , the gap being exactly $\log_2 W_{\text{max}}$. This mirrors item 1 of Remark 4.19, where the same W_{max} loss appears on the knowledge-soundness side; a target of $\lambda = 128$ in the rejection sampling would restore both.

4.4. The accumulator-free variant. In Protocol 12 the integer k needs to handle two unrelated roles: it counts the accumulator pairs the prover samples at Step 1, and it is the number of pieces Π'_{DEC} outputs. These two roles are tied together by our desire to provide a blueprint for a zero-knowledge folding scheme, as outlined in Protocol 13; the sketched protocol $\Pi_{\text{LBF-Fold}}$ aims to feed round r 's output back as round $r+1$'s optional input, and Π_{SuperNeo} 's own Π_{RLC} likewise folds K fresh instances against k accumulators. However, neither duty is a hiding one and the first can be dropped outright.

Indeed, hiding in Appendix B.4 is carried by the Gaussian mask $\mathbf{y}$ of Protocol 7 and, for c_1 , by the blinding block of Definition 3.1. The k sampled pairs enter the simulator of Protocol 19 only as data $\mathcal{S}$ generates itself with the honest sampler and on which it then runs the *honest* prover; none of the hybrid simulators H_0 – H_6 charges anything for them, and they are absent from $N_{\text{PoK}}^{\text{zk}} = 4 + 2t$ for exactly that reason.

Write k_{in} for the number of sampled input accumulators, so that Protocol 12 is the case $k_{\text{in}} = k$, and let Π_{LBF}° be the protocol obtained at $k_{\text{in}} = 0$: Step 1 is deleted, Π'_{R1CS} and Π'_{RLC} run on the single fresh pair alone, so that $\mathbf{v} = \rho_1 \mathbf{z}_1$ and $\mathbf{z} = \mathbf{v} + \mathbf{y}$, while Π'_{DEC} still outputs k pieces.

Corollary 4.24 (Accumulator-free LatticeBlindFold). *Let $K = 1$ and $k_{\text{in}} = 0$, and assume the hypotheses of Theorem 4.13. Then*

$$\Pi_{\text{LBF}}^\circ = \Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}} : \text{CCS}^\circ(b) \longrightarrow \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$$

is a reduction of knowledge and is blinding in the sense of Definition 2.11. Every clause of Theorem 4.13 holds verbatim under the substitution $K + k \mapsto K$ in Propositions 4.15 and 4.17; explicitly,

- (1) *completeness error $\varepsilon_{\text{c}}^{\text{LBF}} \leq (7 + 2t) 2^{-\lambda}$, and $9 \cdot 2^{-\lambda}$ under the batched default of §4.1.2.1;*
- (2) *knowledge-soundness error given by Equation (4.15) with $\varepsilon_{\text{cw}} = W_{\text{max}}/|\mathcal{C}|$ and $N_{\text{PoK}}^{(1)} = 5 + 2t$;*
- (3) *blinding at the same $\varepsilon_{\text{LBF-blind}}$ of Equation (4.6), with $N_{\text{PoK}}^{\text{zk}} = 4 + 2t$ unchanged.*

Moreover the decomposition depth is the least k satisfying

$$(4.20) \quad \tau_{\lambda, \xi} \xi K T(\mathcal{C}) (b - 1) \sqrt{n_{\mathbb{F}}} < b^k,$$

an inequality in which k no longer occurs on the left-hand side.

Proof. Lemmas 4.9, 4.10 and 4.12 are stated for an arbitrary number $K + k_{\text{in}}$ of input pairs and nowhere require $k_{\text{in}} > 0$; in particular the coordinate-wise extractor of Lemma 2.19 is applied with $\ell = K + k_{\text{in}} = 1$, i.e. two accepting branches with $\rho_1 \neq \rho'_1$, and the inversion by $(\rho_1 - \rho'_1)$ is the same one. The error expressions then follow by substitution, the only k_{in} -dependent counts being ε_{cw} and the $(K + k_{\text{in}} + 1)t$ salt-recovering invocations of Steps 1 and 17 of Protocol 7.

For blinding, always considering $K = 1$, delete Step 2 of Protocol 19 and the corresponding summands from $\mathbf{v}^{(\tau)}$, $\mathbf{c}_{\mathbf{y}}$ and $\mathbf{x}_{\mathbf{y}}$. Hybrids H_0 – H_6 go through unchanged: the sampled pairs occurred in them only as objects identically distributed in the real and simulated worlds, so removing them removes no advantage term. In H_4 one still has $\|\mathbf{v}\|_\infty \leq K T(\mathcal{C})(b - 1)$, hence $\mathbf{v} \in V$ and Lemma 3.20 applies at the smaller $L := K T(\mathcal{C})(b - 1)$, which is Equation (4.20). Equation (4.6) is therefore unchanged. ■

The gain is entirely in the parameters. At $K = b - 1 = 1$ the width of Lemma 3.20 drops from $L = (1 + k)T(\mathcal{C})$ to $L = T(\mathcal{C})$, a factor $1 + k \approx 2^5$, i.e. five bits off $\log_2(\tau_{\lambda, \xi} \xi L \sqrt{n_{\mathbb{F}}})$. Re-reading the table of §3.6 at $n_{\mathbb{F}} = 2^{21}$ gives, in place of $k \in \{29, 30, 31\}$,

- $k = 24$ for $T = 128$, $d = 64$, $\xi \approx 13$ (against 29),
- $k = 25$ for $T = 128$, $d = 64$, $\xi \approx 20$ (against 30),
- $k = 25$ for $T = 256$, $d = 128$, $\xi \approx 13$ (against 30),
- $k = 26$ for $T = 256$, $d = 128$, $\xi \approx 20$ (against 31).

Three consequences follow, each proportional to where k or k_{in} appears. First, the public part of the sampled pairs vanishes from the transcript: the ≈ 1.15 MB at $d = 128$ (resp. ≈ 0.80 MB at $d = 64$) that §4.3.3.1 charges to the two “sampled input” rows is removed outright, together with the $k_{\text{in}}t$ ABDLOP commitments and the one transmitted instance $\mathbf{x}^{\text{blind}}$. Second, every remaining k -indexed family shrinks by the ratio $26/31$ (resp. $25/30$), bringing a folding step to roughly 6 MB at $d = 128$ and 4 MB at $d = 64$ on the same accounting. Third, on the soundness side $\varepsilon_{\text{cw}} = W_{\text{max}}(K + k_{\text{in}})/|\mathcal{C}|$ improves from $\approx 2^{-116}$ to 2^{-120} at $d = 64$, so the term that item 1 of Remark 4.19 identifies as limiting that instantiation to ≈ 116 bits is no longer below $\lambda = 120$.

Remark 4.25 (Why the pairs are kept in Protocol 12). The pairs are kept in Protocol 12 for interface compatibility and by our desire to provide a blueprint for a zero-knowledge folding scheme, as outlined in Protocol 13. Π_{LBF}° still outputs an element of $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$, but its input relation is $\text{CCS}^\circ(b)$ alone and has no slot for an incoming accumulator. The second branch of Protocol 13 is therefore unavailable to it, it cannot be iterated, and it cannot be substituted for Π_{RLC} inside a Π_{SuperNeo} pipeline, whose input type is $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^{K+k}$. What Π_{LBF} buys with its k sampled pairs is precisely a step whose input and output types match, at the cost of the five decomposition digits above.

For the one-shot use case this paper is primarily aimed at, namely wrapping a single R1CS circuit in zero-knowledge (§1.2.1), Π_{LBF}° is strictly the better object and the figures of this section are the operative ones. For the blueprint of §4.2.1, the sampled pairs are what make the composition type-correct. All concrete figures reported elsewhere in the paper (§4.3.3.1, §4.3.4) are stated at $k_{\text{in}} = k$, i.e. for Π_{LBF} .

4.5. A comment about Fiat–Shamir. Every figure in §4.3.4 is an *interactive* one, and the interactive statements are the ones proved in this paper. We record here what changes non-interactively, writing Q for the bound on the number of random-oracle queries. The change is not cosmetic: at $q = 2^{64} - 59$ the quadratic extension $\mathbb{K} = \mathbb{F}_{q^2}$ used throughout does *not* suffice at any Q of cryptographic interest, and the deployment path requires $\mathbb{K} = \mathbb{F}_{q^4}$. We state this here rather than in §5 because it is a requirement on the parameters, not an open problem.

Q counts random-oracle queries, not interactions, and it is the reason a non-interactive figure is not the interactive one: a non-interactive proof is produced offline, so nobody rate-limits the prover, which may rehash as often as its budget allows. Two conventions are defensible. Under the *conservative* one, λ bits against adversaries running in time 2^λ means $\log_2 Q = \lambda$, so the interactive errors must be $\approx 2^{-2\lambda}$; this is the usual square-root phenomenon and the reading we recommend for a deployment target. Under the *budget* convention one fixes Q by a plausible hashing budget, in which case a public proof system should still not assume $Q \leq 2^{64}$. We report every requirement as a function of $\lambda + \log_2 Q$ so that either can be read off, and take $Q = 2^{80}$ as the default point. One caveat cuts the other way and we record it without relying on it: a query here is not one hash, since a fresh $\vec{p}$ costs a fresh ABDLOP mask commitment and a re-ground Sum-Check challenge costs a round polynomial, which buys back perhaps 20–30 bits.

Assume the round-by-round property discussed below, so that the loss is a multiplicative $\mathcal{O}(Q)$ on each knowledge-soundness term. Evaluating at the $d = 128$ set ($k = 31, t = 3, K = 1, \log_2 m = 21, D_{\text{max}} = 4$) with $\lambda = 120$ and $Q = 2^{80}$:

term	interactive	requirement for λ bits under FS	at $\lambda + \log_2 Q = 200$
$\varepsilon_{\text{SC}} = \log_2(m) D_{\text{max}} / \mathbb{K} $	$2^{-121.6}$	$ \mathbb{K} \geq \log_2(m) D_{\text{max}} 2^{\lambda + \log_2 Q}$	$ \mathbb{K} \geq 2^{206.4}$
ε_{SZ} (tensored challenges)	$2^{-123.5}$	$ \mathbb{K} \geq (\log_2(m) + 2) 2^{\lambda + \log_2 Q}$	$ \mathbb{K} \geq 2^{204.5}$
$\varepsilon_{\text{CW}} = Q(K + k) / \mathcal{C} $	2^{-243}	$ \mathcal{C} \geq (K + k) 2^{\lambda + \log_2 Q}$	$ \mathcal{C} \geq 2^{205}$
$N_{\text{PoK}}^{(1)} \varepsilon_{\text{ABDLOP}}^{(1)}$	$2^{-140.9}$	$\varepsilon_{\text{ABDLOP}}^{(1)} \leq 2^{-\lambda - \log_2 Q} / N_{\text{PoK}}^{(1)}$	$\varepsilon_{\text{ABDLOP}}^{(1)} \leq 2^{-206.7}$

Three consequences follow, in decreasing order of what they cost to fix. *The extension degree must rise:* at $q = 2^{64} - 59$, $|\mathbb{F}_{q^2}| = 2^{128}$ falls short by some 78 bits and $|\mathbb{F}_{q^3}| = 2^{192}$ by 14.4, sufficing only for $\log_2 Q \leq 65.6$, whereas $|\mathbb{F}_{q^4}| = 2^{256}$ clears the requirement with 50 bits to spare and clears even the conservative convention, which asks for $2^{246.4}$. Remark 3.17 records that §3.3 generalises to $e = 4$ without new hypotheses; at fixed ℓ the packed block counts, and hence the proof size, scale by $e/2 = 2$, which is the honest price of a non-interactive instantiation at these parameters. *$|\mathcal{C}|$ is already large enough at $d = 128$ and is not at $d = 64$:* with $|\mathcal{C}| = 4^d$ one has 2^{256} against the required 2^{205} at $d = 128$, but only 2^{128} at $d = 64$, which is what rules that set out non-interactively (Remark 4.26). *$|\mathcal{C}_{\text{ABDLOP}}|$ must be widened:* the estimator returns $\varepsilon_{\text{ABDLOP}}^{(1)} \approx 2 / |\mathcal{C}_{\text{ABDLOP}}|$ at challenge coefficients bounded by 2, against a requirement of $|\mathcal{C}_{\text{ABDLOP}}| \gtrsim 2^{208}$; since $\mathcal{C}_{\text{ABDLOP}}$ has roughly $d/2 + 1 = 65$ free coefficients at $d = 128$, raising that bound from 2 to about 7 suffices. This last is a knob rather than an obstruction, but it widens η and hence $\mathfrak{s}_1, \mathfrak{s}_2$ and M_{PoK} , and the whole set must be re-run through the estimator; we have not done so.

For a multi-round public-coin protocol the Fiat–Shamir loss is not additive in general, and Π_{LBF} has $\log_2(m) = 21$ Sum-Check rounds, the challenges $\alpha, \gamma, \zeta, \vec{p}, \vec{\mu}$, and the internal challenge rounds of N_{PoK} ABDLOP invocations; the generic multi-round bound would give a vacuous Q^μ degradation. The table above assumes instead the $\mathcal{O}(Q)$ loss that round-by-round soundness would deliver, and we record this

as an *assumption* rather than a theorem: one must prove that each sub-protocol invoked is round-by-round (equivalently, state-restoration) knowledge sound. We have not verified this for the Sum-Check-plus-ABDLOP composition, and regard it as the main gap between the interactive statement proved here and the non-interactive target (item (2) of §5); see Remark B.10 for the separate question of the *simulator* under Fiat–Shamir.

Remark 4.26 (Grinding on the mask loop under Fiat–Shamir). *One term degrades in a way specific to this protocol. The coordinate-wise extraction loss $\varepsilon_{\text{cw}} = W_{\text{max}}(K + k)/|\mathcal{C}|$ of Proposition 4.17 is inflated by W_{max} precisely because W_{max} bounds the number of challenge blocks $\vec{\rho}$ a malicious prover may select among before completing an attempt (Remark 4.3). Interactively this is enforced by $\mathcal{V}$, which resamples $\vec{\rho}$ afresh each iteration; under Fiat–Shamir there is no verifier to enforce it and a prover may rehash freely, so W_{max} must be replaced by Q , giving $\varepsilon_{\text{cw}}^{\text{FS}} = Q(K + k)/|\mathcal{C}|$.*

This is immaterial at $d = 128$, where $|\mathcal{C}| \approx 2^{256}$ gives $\approx 2^{-171}$ at $Q = 2^{80}$, and fatal at $d = 64$, where $|\mathcal{C}| \approx 2^{128}$ gives $\approx 2^{-43}$. The $d = 64$ instantiation, already only ≈ 116 bits interactively (item 1 of Remark 4.19), is therefore not viable non-interactively at any query bound of cryptographic interest; the remedy is again a larger $\mathcal{C}$, at $|\mathcal{C}| \geq (K + k)2^{\lambda + \log_2 Q}$ rather than 2^λ . This cuts against the observation in Remark 2.12 that the attempt caps may be lifted under Fiat–Shamir: lifting them is free for completeness, but the cap’s role in ε_{cw} is a knowledge-soundness one, and there it is replaced by Q , not removed.

5. FUTURE WORK AND OPEN RESEARCH DIRECTIONS

As mentioned in §4.3.3, we expect LatticeBlindFold to admit substantially better performance under further optimization. We collect below the theoretical frameworks and concrete optimizations we regard as most promising, in decreasing order of expected impact, and leave comparing and benchmarking them to future work.

- (1) *Abort-free masking via a Hint-MLWE argument.* The work [KLSS23] drops rejection sampling entirely, masking at a width only a small constant times $\|\text{secret}\|_2$ and arguing simulatability computationally, on the strength of a reduction from Hint-MLWE to standard MLWE. Adopting it here would remove the attempt budgets altogether, and with them both the completeness error of Proposition 4.15 and the $\sqrt{n_{\mathbb{F}}}$ inflation of the norm bound that forces $k = \Theta(\log n_{\mathbb{F}})$ (Remark 4.18). Since that inflation is the dominant structural cost of blinding, this is the single change most likely to close the gap with Π_{SuperNeo} , and we regard it as the most promising direction the paper leaves open.
- (2) *A decider for the output claims.* As recorded in §4.3.3, Π_{LBF} stops at k committed evaluation claims plus their ABDLOP openings, and a deployable zero-knowledge wrapper needs something that checks them succinctly. The natural candidate is to arithmetize the ABDLOP verification, since its linear part is already R1CS-shaped over $\mathbb{R}_{\mathbb{F}}$ and the componentwise instantiation of §3.3 keeps it over the base ring; the obstruction is the norm checks, which want the exact ℓ_2 certificate of [LNP22, §5.2] rather than the relaxed bounds a single invocation gets away with. Solving it would also unblock the recursive branch of Protocol 13, the two needing the same certificate (Remark 4.8), and we regard it as the main missing piece between this paper and a deployable implementation.
- (3) *Round-by-round knowledge soundness: the paper’s main open proof obligation.* The non-interactive figures of §4.5 all assume the $\mathcal{O}(Q)$ Fiat–Shamir loss that round-by-round soundness would buy; with $\log_2(m) = 21$ Sum-Check rounds plus the internal challenges of $N_{\text{PoK}} \approx 8$ batched [LNP22] invocations, the generic multi-round bound is vacuous, so this is a theorem to prove, not a parameter to tune. It must be established for the *composition*, including the W_{max} -capped mask loop of Protocol 7, whose cap has no non-interactive enforcer (Remark 4.26). The simulator side is a separate obligation, Remark B.10 being a ROM statement obtained by programming hash per attempt. By contrast the extension degree is a parameter choice: the quartic set should be produced and estimated with the implementation now in progress, ℓ re-tuned as Remark 3.17 suggests and $\mathcal{C}_{\text{ABDLOP}}$ widened accordingly. Until then the non-interactive figures are conditional and the interactive ones are the results of this paper.
- (4) *Report core-SVP alongside the root Hermite factor.* We follow [LNP22, §6.1] in measuring lattice hardness by the root Hermite factor $\delta < 1.0045$, which keeps our figures comparable with theirs but is a coarse proxy at the margins we claim. A current estimator should be run on the three

instances that matter, namely the Ajtai MSIS at $4TB$, the ABDLOP MSIS at the extracted bound of Lemma 3.6 evaluated at the widest call, and the MLWE instance, with core-SVP costs reported for each; binding at β_{rlx} needs no run of its own, being dominated by the second.

- (5) *Defer and batch every proof of knowledge.* Soundness is anchored by the commitments, which are sent at the right times, so the proofs of knowledge about them can all be deferred and batched in the style of [LNP22]. That would take ≈ 7 invocations down to 2, cutting both the round count and the additive $\kappa_{\text{com}}(m_1 + m_2)d$ communication terms roughly threefold, at the mild price of a $\sqrt{N_c}$ growth in the batched masking width. Combined with the deduplication of Remark 4.14, this is the most direct saving available without changing the protocol’s structure.
- (6) *The decomposition basis as a knob, and a parameter sweep.* Taking $b = 4$ halves k while doubling the Sum-Check degree, which is roughly a wash on the prover since $k \cdot D_{\text{max}}$ is invariant, but halves the k -multiplied ABDLOP, proof-of-knowledge and communication terms, and ktd in ε_{SZ} with them. More generally, once the implementation now in progress is available, the illustrative points should be replaced by a sweep over $(b, d, \xi, n_{\mathbb{F}}, q, T(\mathcal{C}))$; several choices we made for analytical convenience, such as $\xi \approx 20$ and a quadratic $\mathbb{K}$, are plausibly not optimal.
- (7) *Higher-degree extensions, and binary fields.* Given the recent interest in towers of binary fields, beginning with [DP23] and continuing with [Set26, BRW26], it would be interesting to investigate protocols of this shape over extensions of much higher degree than the quadratic $\mathbb{K} = \mathbb{F}_{q^2}$ we use. Several of our choices are degree-dependent, among them the componentwise instantiation of §3.3, the single execution of $\Pi_{\text{many}}^{(\text{ct})}$, and the $\varepsilon_{\text{SZ}}, \varepsilon_{\text{SC}}$ terms, so the trade-off is not simply a matter of enlarging $|\mathbb{K}|$; and the lattice layer has no characteristic-2 analogue that we know of.

Several narrower questions are left open along the way: whether blinding extends to $K \geq 2$, whether a Lova-style analogue can be built over $\mathbb{Z}/2^{64}\mathbb{Z}$, and whether the rejection-sampling requirements can be relaxed to admit smaller primes. In the end, we plan to implement the protocol and include detailed benchmarks.

ACKNOWLEDGEMENTS. Many thanks to Wyatt Benno and Alberto Centelles (in alphabetical order) for suggesting this line of research, and for their insightful comments and advice throughout the work, which led to significant improvements and resolved key issues. Moreover, the author is grateful to Antoine Douchet, Khalil Gibran, Sebastien Guillemot, and Srinath Setty (in alphabetical order) for several fruitful discussions.

REFERENCES

- [ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009. URL: <https://www.iacr.org/archive/crypto2009/56770585/56770585.pdf>, doi:10.1007/978-3-642-03356-8_35. 21
- [AL21] Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings: Limits of schnorr-like arguments over lattices. Cryptology ePrint Archive, Paper 2021/202, 2021. URL: <https://eprint.iacr.org/2021/202.14>
- [APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Paper 2015/046, 2015. GitHub repository <https://github.com/malb/lattice-estimator>. URL: <https://eprint.iacr.org/2015/046.53>
- [AST23] Arasu Arun, Srinath Setty, and Justin Thaler. Jolt: SNARKs for virtual machines via lookups. Cryptology ePrint Archive, Paper 2023/1217, 2023. URL: <https://eprint.iacr.org/2023/1217.3,5>
- [Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–636, 1993. URL: <http://eudml.org/doc/165105.32>
- [BC24] Dan Boneh and Binyi Chen. LatticeFold: A lattice-based folding scheme and its applications to succinct proof systems. Cryptology ePrint Archive, Paper 2024/257, 2024. URL: <https://eprint.iacr.org/2024/257.1,3,19>
- [BC25] Dan Boneh and Binyi Chen. LatticeFold+: Faster, simpler, shorter lattice-based folding for succinct proof systems. Cryptology ePrint Archive, Paper 2025/247, 2025. URL: <https://eprint.iacr.org/2025/247.1,3>
- [BCDG26] Wyatt Benno, Alberto Centelles, Antoine Douchet, and Khalil Gibran. Jolt atlas: Verifiable inference via lookup arguments in zero knowledge, 2026. URL: <https://arxiv.org/abs/2602.17452>, arXiv:2602.17452. 1, 3, 5
- [BCS23] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Lattice-based succinct arguments for NP with polylogarithmic-time verification. Cryptology ePrint Archive, Paper 2023/930, 2023. URL: <https://eprint.iacr.org/2023/930.32>

- [BDL⁺16] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. Cryptology ePrint Archive, Paper 2016/997, 2016. URL: <https://eprint.iacr.org/2016/997>. 1, 4, 6, 8, 18, 19, 21, 39
- [BRW26] Benedikt Bünz, Ron Rothblum, and William Wang. Flock: Fast proving for batch boolean computations. Cryptology ePrint Archive, Paper 2026/1329, 2026. URL: <https://eprint.iacr.org/2026/1329>. 60
- [BZG⁺26] Ryan Babbush, Adam Zalcman, Craig Gidney, Michael Broughton, Tanuj Khattar, Hartmut Neven, Thiago Bergamaschi, Justin Drake, and Dan Boneh. Securing elliptic curve cryptocurrencies against quantum vulnerabilities: Resource estimates and mitigations, 2026. URL: <https://arxiv.org/abs/2603.28846>, [arXiv:2603.28846](https://arxiv.org/abs/2603.28846). 4
- [CFW26] Alessandro Chiesa, Giacomo Fenzi, and Guy Weissenberg. Zero-knowledge IOPs for constrained interleaved codes. Cryptology ePrint Archive, Paper 2026/391, 2026. URL: <https://eprint.iacr.org/2026/391>. 31
- [DHRR26] Rahul Dalal, Tamir Hemo, Eugene Rabinovich, and Ron D. Rothblum. Veil: Lightweight zero-knowledge for hash-based multilinear proof systems. *GitHub*, 23 March, 2026. URL: <https://github.com/succinctlabs/sp1/blob/main/slop/crates/veil/paper/veil.pdf>. 3
- [DP23] Benjamin E. Diamond and Jim Posen. Succinct arguments over towers of binary fields. Cryptology ePrint Archive, Paper 2023/1784, 2023. URL: <https://eprint.iacr.org/2023/1784>, doi:10.1007/978-3-031-91134-7_4. 60
- [FKNP24] Giacomo Fenzi, Christian Knabenhans, Ngoc Khanh Nguyen, and Duc Tu Pham. Lova: Lattice-based folding scheme from unstructured lattices. Cryptology ePrint Archive, Paper 2024/1964, 2024. URL: <https://eprint.iacr.org/2024/1964>. 4
- [FMN23] Giacomo Fenzi, Hossein Moghaddas, and Ngoc Khanh Nguyen. Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency. Cryptology ePrint Archive, Paper 2023/846, 2023. URL: <https://eprint.iacr.org/2023/846>. 15, 32, 65, 66
- [Fou25] LF Decentralized Trust (The Linux Foundation). Nightstream: A lattice zkvm (post-quantum zkvm), August 2025. Accessed: 2026-07-22. URL: <https://github.com/LFDT-Nightstream/Nightstream>. 53
- [GLLO26] Albert Garreta, Helger Lipmaa, Urmas Luhaäär, and Michał Osadnik. Cyclo: Lightweight lattice-based folding via partial range checks. Cryptology ePrint Archive, Paper 2026/359, 2026. URL: <https://eprint.iacr.org/2026/359>. 1, 3
- [KLOT25] Shuto Kuriyama, Russell W. F. Lai, Michał Osadnik, and Lorenzo Tucci. SALSAA – sumcheck-aided lattice-based succinct arguments and applications. Cryptology ePrint Archive, Paper 2025/2124, 2025. URL: <https://eprint.iacr.org/2025/2124>. 4
- [KLSS23] Duhyeon Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. Cryptology ePrint Archive, Paper 2023/623, 2023. URL: <https://eprint.iacr.org/2023/623>, doi:10.1007/978-3-031-38554-4_18. 59
- [KP22] Abhiram Kothapalli and Bryan Parno. Algebraic reductions of knowledge. Cryptology ePrint Archive, Paper 2022/009, 2022. URL: <https://eprint.iacr.org/2022/009>. 13
- [KS23] Abhiram Kothapalli and Srinath Setty. HyperNova: Recursive arguments for customizable constraint systems. Cryptology ePrint Archive, Paper 2023/573, 2023. URL: <https://eprint.iacr.org/2023/573>. 1, 3, 13
- [KS25] Darya Kaviani and Srinath Setty. Vega: Low-latency zero-knowledge proofs over existing credentials. Cryptology ePrint Archive, Paper 2025/2094, 2025. URL: <https://eprint.iacr.org/2025/2094>. 1, 3
- [KST22] Abhiram Kothapalli, Srinath Setty, and Ioanna Tzialla. Nova: Recursive zero-knowledge arguments from folding schemes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 359–388, Cham, 2022. Springer Nature Switzerland. 3
- [LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. Cryptology ePrint Archive, Paper 2022/284, 2022. URL: <https://eprint.iacr.org/2022/284>. 1, 4, 6, 8, 9, 10, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 32, 33, 39, 47, 53, 54, 55, 59, 60, 63, 64, 65, 70
- [LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. Cryptology ePrint Archive, Paper 2020/1448, 2020. URL: <https://eprint.iacr.org/2020/1448>. 1, 19, 20
- [LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015. doi:10.1007/s10623-014-9938-4. 19, 21
- [LS17] Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. Cryptology ePrint Archive, Paper 2017/523, 2017. URL: <https://eprint.iacr.org/2017/523>. 14
- [Mic16] Daniele Micciancio. The gaussians distribution. Lecture notes, CSE 206A: Lattice Algorithms and Applications, 2016. Winter 2016. URL: <https://cseweb.ucsd.edu/classes/fa17/cse206A-a/LecGaussian.pdf>. 32
- [NOZ26] Ngoc Khanh Nguyen, George O’Rourke, and Jiapeng Zhang. Hachi: Efficient lattice-based multilinear polynomial commitments over extension fields. Cryptology ePrint Archive, Paper 2026/156, 2026. URL: <https://eprint.iacr.org/2026/156>. 31
- [NS24] Ngoc Khanh Nguyen and Gregor Seiler. Greyhound: Fast polynomial commitments from lattices. Cryptology ePrint Archive, Paper 2024/1293, 2024. URL: <https://eprint.iacr.org/2024/1293>, doi:10.1007/978-3-031-68403-6_8. 18, 31
- [NS26] Wilson Nguyen and Srinath Setty. Neo and SuperNeo: Post-quantum folding with pay-per-bit costs over small fields. Cryptology ePrint Archive, Paper 2026/242, 2026. URL: <https://eprint.iacr.org/2026/242>. 1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 39, 47, 48, 49, 50, 51, 53, 62, 63, 64, 65, 66, 67, 68, 69, 70

- [Osa26] Michał Osadnik. PikkuFold: Efficient folding in a few kilobytes. Cryptology ePrint Archive, Paper 2026/1809, 2026. URL: <https://eprint.iacr.org/2026/1809>. 4
- [Set26] Srinath Setty. BinarySpartan: Spartan over binary fields. Cryptology ePrint Archive, Paper 2026/1656, 2026. URL: <https://eprint.iacr.org/2026/1656>. 60
- [Tha22] Justin Thaler. Proofs, arguments, and zero-knowledge. *Foundations and Trends® in Privacy and Security*, 4(2–4):117–660, 2022. Updated version also available at <https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.pdf>. doi:10.1561/33000000030. 30, 31, 62
- [The25] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2025. DOI 10.5281/zenodo.14600115. URL: <https://www.sagemath.org>. 53
- [Was97] Lawrence C. Washington. *Introduction to cyclotomic fields.*, volume 83 of *Grad. Texts Math.* New York, NY: Springer, 2nd ed. edition, 1997. 32
- [XZZ⁺19] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. Cryptology ePrint Archive, Paper 2019/317, 2019. URL: <https://eprint.iacr.org/2019/317>. 5, 30, 31, 39, 62, 72

APPENDIX A. AI USAGE DISCLOSURE

Portions of this manuscript were edited with the assistance of an AI writing tool (Claude AI and, in minimal part, Gemini AI), which was used to improve grammar, wording, and formatting consistency, as well as to adjust auxiliary algorithms, help with the complexity analyses, audit previous versions of the protocol, and suggest and implement improvements. The tool was used for drafting and for cross-checking. All technical content (including definitions, theorems, and proofs) originates with, and has been verified by, the author, who takes full responsibility for the correctness and originality of the work.

APPENDIX B. DEFERRED PROOFS

B.1. Proof of Lemma 4.9.

We begin by showing that the protocol is complete, then that it is public coin, and in the end we show the properties of strong interactive reduction.

Completeness: The Sum-Check protocol is perfectly complete, while the proof of knowledge protocol in Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$, the wrapper $\Pi_{\text{many}}^{(\text{ct})}$ (Lemma 3.8), the quadratic protocol $\Pi_{\text{ABDLOP, many}}^{(2)}$ (Lemma 3.7) and Π^{anc} (Lemma 3.16) are each complete with error $2^{-\lambda}$ (by assumption). Since the protocol makes five such invocations, and each fails independently (the retry loops use independent masks, and no commitment is resampled between them), a union bound gives $\varepsilon_c^{\text{R1CS}} \leq 5 \cdot 2^{-\lambda}$. Recall that Lemma 2.24 proves that the CCS reduction of SuperNeo is complete, therefore, we must take care of parts introduced in Protocol 6. The masking procedure does not alter the completeness of the protocol (cf. [Tha22, XZZ⁺19]). The batched $\mathbb{Z}_q$ -linear proof of knowledge certifying the degree-0 surrogates $\{\hat{y}_{i,j}\}$ (Step 10) is honestly satisfied by construction: $\text{cf}(\hat{y}_{i,j})_\ell = 0$ for $\ell \geq 2$ by sampling, and $\text{ct}(\hat{y}_{i,j}) = \text{ct}(y'_{i,j})$ by definition, so both relations of Step 12 hold with π_μ -probability 1 over the honest transcript. By construction of Commit(F), Commit(N), Commit(E) (from $\hat{y}_{i,j}$, cf. Lemma 2.20), the fact that the ABDLOP commitment scheme contains the message in plain, and Lemma 2.31, the honestly-computed opening satisfies $\text{ct}(u) = u^*$ for $u^* := h - \text{eq}(r', \alpha) \cdot (c_F + \delta_0 \cdot c_N) - \delta_1 \cdot c_E$, i.e. $u^* \stackrel{?}{=} \zeta \cdot S_{2,p}$ (a single salt-block identity on $\mathbf{s}_{P,2}$, cf. the weight table of §3.2.1.1, so completeness of Π^{anc} (Lemma 3.16) carries this check rather than an independently-stated equation.

Public Coin: The Sum-Check protocol, the proof of knowledge protocol in Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$, as well as Protocol 4 are public-coin and all other verifier’s messages $\alpha \xleftarrow{\$} \mathbb{K}^{\log_2(m)}$, $\gamma \xleftarrow{\$} \mathbb{K}$, $(\delta_0, \delta_1) \xleftarrow{\$} \mathbb{K}^2$, $\zeta \xleftarrow{\$} \mathbb{K}^\times$, and $r' \xleftarrow{\$} \mathbb{K}^{\log_2(m)}$ are uniformly sampled from their corresponding set.

Strong Interactive Reduction (1st property): Since the function ϕ we consider is the same of the unmodified Π_{CCS} reduction of [NS26, §7.3] and the commitments $\{c_i\}_{i \in [K+k]}$ are handled exactly as in this protocol, we deduce that the property is satisfied as in [NS26, Appendix B.2].

Strong Interactive Reduction (2nd property): We begin by fixing the relations required in Definition 2.7 as $\mathcal{R}_1 = \text{CCS}(b, \mathcal{L})^K \times \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$, $\mathcal{R}_2 = \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^{K+k}$, and $\mathcal{R}'_2 = \text{CE}_{\text{com}}^{\text{rlx},3}(q/2, \mathcal{L}, \beta_{\text{rlx}}^{(3)}, \text{Commit})^{K+k}$ with $\mathcal{R}_2 \subset \mathcal{R}'_2$. Now, we consider an arbitrary EPT adversary $(\mathcal{A}, \mathcal{P}^*)$, such that the relaxed success probability (recall Definition 2.7.(2)) of the adversary $\epsilon'(\mathcal{A}, \mathcal{P}^*) \geq 1/\text{poly}(\lambda)$ and

$$\Pr \left[\begin{array}{c} w_2, w'_2 \neq \perp \\ \wedge \\ w_2 \neq w'_2 \end{array} \middle| \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ (u_2, w_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}) \\ (u'_2, w'_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}) \end{array} \right] \leq \text{negl}(\lambda).$$

As in [NS26, Appendix B.2], we construct an EPT extractor $\mathcal{E}$ such that

$$\Pr \left[(\text{pp}, \text{s}, u_1, w_1) \in \mathcal{R}_1 \middle| \begin{array}{l} \text{pp} \leftarrow \mathcal{G}(1^\lambda, \text{sz}) \\ (\text{s}, u_1, \text{st}) \leftarrow \mathcal{A}(\text{pp}) \\ (\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s}) \\ w_1 \leftarrow \mathcal{E}(\text{pp}, \text{s}, u_1, \text{st}) \end{array} \right] \geq \epsilon'(\mathcal{A}, \mathcal{P}^*) - \text{negl}(\lambda),$$

as in Protocol 14. Note that we need to modify slightly the extractor presented in [NS26, Appendix B.2] as our relations $\mathcal{R}_2$ and $\mathcal{R}'_2$ include the ring vectors $y_{i,j}$, $y'_{i,j}$ in the witness. We highlight these modifications in green.

Protocol 14 $\mathcal{E}(\text{pp}, \text{s}, u_1, \text{st}) \rightarrow w_1$:

```

1:  $(\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \text{s})$ .
2: Assign  $\text{result} := \perp$ .
3: while  $\text{result} = \perp$  do
4:   Simulate  $\text{result} \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}^*)$ 
5:   if  $\text{result} \neq \perp$  then
6:     Parse  $(u_2, w_2) \leftarrow \text{result}$ .
7:     if  $(\text{pp}, \text{s}, u_2, w_2) \notin \mathcal{R}'_2$  then
8:       Set  $\text{result} = \perp$ .
9:     end if
10:  end if
11: end while
12: Simulate  $\text{result}' \leftarrow \langle \mathcal{P}^*, \mathcal{V} \rangle((\text{pk}, \text{vk}), u_1, \text{st}^*)$ 
13: if  $\text{result}' \neq \perp$  then
14:   Parse  $(u'_2, w'_2) \leftarrow \text{result}'$ .
15:   if  $(\text{pp}, \text{s}, u'_2, w'_2) \notin \mathcal{R}'_2$  then
16:     Set  $\text{result}' = \perp$ .
17:   end if
18: end if
19: if  $\text{result}' = \perp$  then
20:   return  $\perp$ .
21: end if
22: Parse  $(u_2, w_2) \leftarrow \text{result}$  and  $(u'_2, w'_2) \leftarrow \text{result}'$ .
23: if  $\psi(w_2) \neq \psi(w'_2)$  then ▷ i.e. the two runs disagree on some  $\mathbf{z}_i$ .
24:   return  $\perp$ .
25: end if
26: Extract  $(\mathbf{z}_1, \dots, \mathbf{z}_{K+k}) \leftarrow \psi(w_2)$  and read  $r'$  off  $u_2$ .
27: Since  $\mathcal{P}^*$  passes the commitment check at Step 1 of Protocol 6, then we can proceed as in the proof of
    Lemma 3.6 (see [LNP22, Lemma 3.1]) and define a EPT extractor  $\mathcal{E}'$  which extracts the corresponding
    salts.

```

For the degree-0 surrogate commitments $\{\hat{c}_{i,j}\}$ introduced at the (new) linear-relation step, invoke the same extractor $\mathcal{E}'$ once more: since $\mathcal{P}^*$ passes the batched $\mathbb{Z}_q$ -linear ABDLOP proof of knowledge for $\{\text{ct}(\pi_\mu \cdot \hat{y}_{i,j}) = 0, \text{ct}(y'_{i,j} - \hat{y}_{i,j}) = 0\}$, the extractor recovers openings $(\tilde{y}_{i,j}, \tilde{y}'_{i,j})$ which, by Lemma 3.13, satisfy $\text{cf}(\tilde{y}_{i,j})_\ell = 0$ for all $\ell \geq 2$ except with probability at most $(2/|\mathbb{K}|)$ over $\{\mu_\ell\}$, and satisfy $\text{ct}(\tilde{y}'_{i,j}) = \text{ct}(\tilde{y}_{i,j})$ by the second relation.

Run the extractor of Lemma 3.8 on the $\Pi_{\text{many}}^{(\text{ct})}$ invocation of Step 12 of Protocol 6 for the family $\{\text{ct}(\pi_\mu \cdot \hat{y}_{i,j}) = 0, \text{ct}(y'_{i,j} - \hat{y}_{i,j}) = 0\}_{i \in [K+k], j \in [t]}$.

Rewind $\mathcal{P}^*$ on the challenge vector $\vec{\gamma}$ of that sub-protocol and apply the extractor of Lemma 3.6 to its inner $\Pi_{\text{many}}^{(1)}$ call, obtaining openings $(\tilde{y}_{i,j}, \tilde{y}'_{i,j}, \tilde{g})$ of the committed messages with $\sum_j \gamma_j f_j(\tilde{s}_1, \tilde{\mathbf{m}}) + \tilde{g} = h$ in $\mathbb{R}_{\mathbb{K}}$. Taking constant coefficients and using $\mathcal{V}$'s check, every $\text{ct}(f_j)$ vanishes except with probability $\varepsilon_{\text{ct}} = 1/|\mathbb{K}|$ over $\vec{\gamma}$.

Discard the auxiliary openings $\tilde{y}_{i,j}, \overline{m_{sq,i}}, \overline{m_{cube,i}}, \overline{m_{prod,i}}$ (they served only to certify the scalar relations); retain $\tilde{y}'_{i,j}$ as the extracted opening of $y'_{i,j}$.

Assign the extracted salts (of $c'_{i,j}, \hat{c}_{i,j}$, and the quadratic commitments) to $\{s_{i,j}\}_{i \in \{K+1, \dots, K+k\}, j \in [t]}$.

28: **for** all $i \in [K]$ **do**

29: Assign $w_i^{\text{CCS}} \leftarrow z_i[n_{\mathbb{F}, \text{in}} :]$ for $i \in [K]$.

30: **end for**

31: **return** $w_1 := (w_1^{\text{CCS}}, \dots, w_K^{\text{CCS}}) \times (z_{K+1}, \dots, z_{K+k}, \{y_{K+1,j}, \dots, y_{K+k,j}\}_{j \in [t]}, \{s_{i,j}\}_{i \in \{K+1, \dots, K+k\}, j \in [t]})$.

Note that, the initial consistency check in Protocol 6 is necessary for obtaining a strong interactive reduction; cf. Remark 2.29. The extractor constructed in the proof of [LNP22, Lemma 3.1] manages to extract a solution to the MSIS problem if the prover $\mathcal{P}^*$ passes with a non-negligible probability. Such probability is contained in the relaxed success probability $\epsilon'(\mathcal{A}, \mathcal{P}^*)$.

Now, since we are enforcing the ZKPoK checks at Steps 7 and 18 (we require them to link the evaluation to the correct execution of the Sum-Check protocol and then call its soundness, cf. Remark 4.2), we can then proceed as in [NS26, Appendix B.2], mutatis mutandis. The resulting knowledge-soundness error is the sum

$$\varepsilon_{\text{R1CS}} \leq \varepsilon_{\text{SZ}} + \varepsilon_{\text{SC}} + \varepsilon_{\zeta} + \varepsilon_{\text{deg } 0} + 3\varepsilon_{\text{ct}} + 4\varepsilon_{\text{ABDLOP}}^{(1)} + \varepsilon_{\text{ABDLOP}}^{(2)},$$

where the first group follows loc. cit. (the Schwartz–Zippel error of the batching challenges $\gamma, \delta_0, \delta_1, \alpha$, the Sum-Check protocol itself, and the blinding challenge ζ), the second group is contributed by our modifications (Lemmas 3.13, 3.8), and the last group by the ABDLOP proofs of knowledge of Steps 1, 7, 12, 15 and 18.

Let us stress that ε_{SZ} is not inherited from SuperNeo. Loc. cit. batches with the powers of a single challenge and separates the three families by further powers γ^K, γ^{2K+k} ; we batch with the tensored challenges of Step 2 and separate by the independent δ_0, δ_1 (Remark 4.2.(4)). Lemma 4.16 gives

$$\varepsilon_{\text{SZ}} \leq \frac{\log_2(m) + 2}{|\mathbb{K}|},$$

in place of the higher bound that the power-based batching would incur.

All terms are spelled out, and the sum is negligible in λ (see Proposition 4.17).

B.2. Proof of Lemma 4.10. We prove here Lemma 4.10. Before diving into the proof we state and prove a useful lemma about the norm of the salts.

Lemma B.1. *For the random linear combination reduction Π'_{RLC} defined in Protocol 7, the salt combinations*

$$s_{1,j} = \sum_{i \in [K+k]} \rho_i s_{1,i,j} + s_{1,\mathbf{y},j} \quad \text{and} \quad s_{2,j} = \sum_{i \in [K+k]} \rho_i s_{2,i,j} + s_{2,\mathbf{y},j}$$

computed at Step 19 of Protocol 7 satisfy, for both $l \in [2]$ and every $j \in [t]$,

$$\|s_{l,j}\|_{\infty} < B_{\text{fold}} := (K+k) \cdot T(\mathcal{C}) \cdot (\tilde{B} - 1) + \tilde{B}.$$

No further hypothesis is required. In particular the conditions $\tau_{\lambda,\xi} \xi (K+k) T(\mathcal{C}) (\tilde{B} - 1) < b^k$ and $B = b^k \leq \tilde{B}^k$ of the decomposing version are no longer needed, since Π'_{DEC} (Protocol 8) resets the salt bound by resampling rather than by $\tilde{B}$ -ary decomposition. The salt budget is thereby decoupled from the witness budget that fixes k .

Proof. Fix $l \in [2]$ and $j \in [t]$. By construction (Protocol 7, cf. also Protocol 9), each individual salt satisfies $\|s_{l,i,j}\|_{\infty} < \tilde{B}$ for $i \in [K+k]$, and likewise the freshly sampled mask salt satisfies $\|s_{l,\mathbf{y},j}\|_{\infty} < \tilde{B}$. By the defining property of the expansion factor $T(\mathcal{C})$ (§2.3), for every $\rho_i \in \mathcal{C}$ and every ring element v we have $\|\rho_i v\|_{\infty} \leq T(\mathcal{C}) \cdot \|v\|_{\infty}$. Applying this to each of the $K+k$ scaled terms and the triangle inequality across the sum,

$$\left\| \sum_{i \in [K+k]} \rho_i s_{l,i,j} \right\|_{\infty} \leq \sum_{i \in [K+k]} \|\rho_i s_{l,i,j}\|_{\infty} \leq \sum_{i \in [K+k]} T(\mathcal{C}) \cdot \|s_{l,i,j}\|_{\infty} < (K+k) \cdot T(\mathcal{C}) \cdot (\tilde{B} - 1).$$

The additional term $s_{l,\mathbf{y},j}$ is, by design, not scaled by any ρ_i as it is folded in with a fixed coefficient of 1, so it contributes to the total bound only its own norm, $\|s_{l,\mathbf{y},j}\|_\infty < \tilde{B}$, with no expansion-factor amplification. One further application of the triangle inequality gives

$$\|s_{l,j}\|_\infty = \left\| \sum_{i \in [K+k]} \rho_i s_{l,i,j} + s_{l,\mathbf{y},j} \right\|_\infty \leq \left\| \sum_{i \in [K+k]} \rho_i s_{l,i,j} \right\|_\infty + \|s_{l,\mathbf{y},j}\|_\infty < (K+k) \cdot T(\mathcal{C}) \cdot (\tilde{B} - 1) + \tilde{B}.$$

The only requirement on B_{fold} is that $\text{MSIS}_{\kappa, m_1+m_2, 2B_{\text{fold}}\sqrt{(m_1+m_2)d}}$ be hard, so that Commit remains binding on folded salts. $\blacksquare$

We can now dive into the proof of the lemma.

We begin by showing that the protocol is complete, then that it is public coin, and in the end we show the properties of weak interactive reduction.

Completeness: By Lemma B.1, Lemma 3.20 and Lemma 2.25, an honest execution that leaves the loop produces an output in the target relation. Two distinct failure modes remain, and they must be charged separately.

The first is the mask loop itself. The algorithm is still PPT and concludes with $\approx 2, 3$ repetitions (of the first sampling lines) in expectation, and the attempt cap of Protocol 7 truncates only the tail of this distribution. More precisely, by Lemma 3.20 and the independence of the masks $\mathbf{y}$ drawn across attempts, an honest prover fails to obtain an accepting attempt within W_{max} iterations with probability at most $(1 - 1/M)^{W_{\text{max}}} \leq 2^{-\lambda}$, exactly as an Rej_1 abort is charged in [LNP22]; cf. Remark 4.3.

The second is the ABDLOP proofs of knowledge, which Protocol $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ performs with abort, each is complete only with error $2^{-\lambda}$, under its own, independent, retry budget $W_{\text{max}}^{\text{PoK}}$. More precisely, Every invocation of an ABDLOP proof of knowledge carries a rejection loop of its own (Protocols $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ and $\Pi_{\text{ABDLOP}}^{(2)}((s_2, s_1, m), \sigma, f)$). We write M_{PoK} for the joint repetition rate of that loop, i.e. the reciprocal of the probability that Rej_1 and Rej_2 both accept in a given attempt, and

$$(B.1) \quad W_{\text{max}}^{\text{PoK}} := \left\lceil \lambda / \log_2(M_{\text{PoK}} / (M_{\text{PoK}} - 1)) \right\rceil$$

for the corresponding attempt budget, fixed publicly once and for all, and enforced by $\mathcal{V}$ exactly as W_{max} is (Remark 4.3). Step 1 makes one such invocation per pair $(i, j) \in [K+k] \times [t]$ and Step 17 one per $j \in [t]$, whence $(K+k)t + t$ of them. These budgets are nested inside the mask loop but do not interact with it. A proof-of-knowledge retry resamples only $(\mathbf{y}_1, \mathbf{y}_2)$ and the challenge c of that sub-protocol, never the mask $\mathbf{y}$, the mask commitments $C_{\mathbf{y},0}, \{C_{\mathbf{y},j}\}$, or the folding challenges $\vec{\rho}$, so no proof-of-knowledge retry can be used to obtain a second draw of $\vec{\rho}$.

A union bound over the $1 + (K+k)t + t$ failure events gives the stated $\varepsilon_c^{\text{RLC}}$.

Public Coin: The proof of knowledge protocol in Protocol $\Pi_{\text{many}}^{(1)}((s_2, s_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ is public coin, and the verifier's randomness consists of challenges $\rho_1, \dots, \rho_{K+k}$, which are sampled uniformly at random from $\mathcal{C}$ and sent to the prover.

Weak Interactive Reduction:

As in [NS26, Appendix B.3] we make use of Lemma 2.19, but before proceeding with it, we must record a couple of important remarks on why we can still apply this lemma.

The mask $\mathbf{y}$ can be treated as fixed under the rewind: the mask commitments for a given attempt are always sent before the corresponding $\rho_1, \dots, \rho_{K+k}$ are sampled, so the protocol's public-coin structure forces any malicious prover to fix its mask commitments before it has any information about the challenges it is about to be rewound on. Below, we make this precise by building the fixed mask prefix explicitly into the state st^* passed to the adversary function $A_{(\text{pp}, s, u_1, \text{st}^*)}$, rather than relying on the single-round formulation of the adversary function in [FMN23, Figure 11], which has no inner rejection loop and so leaves this point implicit. Once the mask is fixed, the witnesses $z_1, \dots, z_{K+k}$ are recovered exactly as in [NS26, Appendix B.3] via Lemma 2.19. More precisely, this holds true as all quantities are now shifted by the same $\mathbf{y}$ and the same salts. The mask $\mathbf{y}$ itself, together with $c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}, \{y_{\mathbf{y},j}\}$, is recovered separately, from the final, accepting transcript; since $\mathcal{P}^*$ passes the commitment check then we can proceed as in the proof of Lemma 3.6 (see [LNP22, Lemma 3.1]) and define a EPT extractor $\mathcal{E}'$ which extracts the corresponding salts.

Following Definition 2.6, we consider an arbitrary EPT adversary $(\mathcal{A}, \mathcal{P}^*)$ for Π'_{RLC} with success probability, $\epsilon(\mathcal{A}, \mathcal{P}^*) \geq 1/\text{poly}(\lambda)$. In order to precisely define the adversary function, we need to modify the single-round setting of [FMN23, Figure 11], since we must handle the Rej_1 while-loop. The adversary and verification functions below are therefore parametrised by a transcript prefix

$$\text{tr}_{\text{pre}} := (C_{\mathbf{y},0}, \{C_{\mathbf{y},j}\}_{j \in [t]})$$

together with $\mathcal{P}^*$'s internal state $\text{st}_{\mathcal{P}^*}$ at the point where that prefix was sent (its random tape, folded in via an averaging argument over EPT provers). We write $\text{st}^* := (\text{st}_{\mathcal{P}^*}, \text{tr}_{\text{pre}})$ and treat it as fixed under all the rewinding that follows: tr_{pre} , and hence $\mathbf{y}$ and its associated salts and commitments, are never resampled. How st^* is *generated* is a separate question, deferred to Protocol 17; we note here only that it is *not* obtained by running $\mathcal{P}^*$ until its first Rej_1 -accepting attempt, for the reason given after that protocol. Lemma 2.19 applies to every fixed st^* alike, so nothing in the present paragraph depends on the choice.

We can now set the machinery of Lemma 2.19 in motion, by constructing an adversary function $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$ and a verification function $V_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$ as in Protocol 15 and Protocol 16 respectively. We highlight in green all changes from [NS26, Appendix B.3].

Protocol 15 $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}(\vec{c})$:

- 1: Execute encoder $(\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \mathbf{s})$.
 - 2: Parse $(\text{st}_{\mathcal{P}^*}, \text{tr}_{\text{pre}}) \leftarrow \text{st}^*$.
 - 3: Resume $\mathcal{P}^*$ from state $\text{st}_{\mathcal{P}^*}$, having already sent tr_{pre} (i.e. its mask commitments $C_{\mathbf{y},0}, \{C_{\mathbf{y},j}\}_{j \in [t]}$, cf. Remark B.9), and simulate $(u_2, w_2) \leftarrow \langle \mathcal{P}^*, \mathcal{V}(\text{vk}, u_1) \rangle$ letting the verifier send $\rho_1, \dots, \rho_{K+k} := \vec{c}$ as the only subsequent messages.
 - 4: **return** w_2
-

Without overburdening the algorithmic notation, we assume that $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}(\vec{c})$ returns $\perp$ if, upon receiving $\vec{c}$, $\mathcal{P}^*$ initiates a further iteration of the loop rather than completing the current attempt. A then answers exactly one challenge block, as Lemma 2.19 requires.

Protocol 16 $V_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}(\vec{c}, w_2) \rightarrow \{0, 1\}$:

- 1: Execute encoder $(\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \mathbf{s})$.
 - 2: Parse $(\text{st}_{\mathcal{P}^*}, \text{tr}_{\text{pre}}) \leftarrow \text{st}^*$ and resume $\mathcal{P}^*$ as in Protocol 15, simulating $(u_2, \_) \leftarrow \langle \mathcal{P}^*, \mathcal{V}(\text{vk}, u_1) \rangle$ with $\rho_1, \dots, \rho_{K+k} := \vec{c}$.
 - 3: **return** 1 (i.e. accept) if and only if $(u_2, w_2) \in \text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit})$. ▷ The relaxed relation only enlarges the accept set, so $\epsilon^{\mathcal{V}}(\mathcal{A})$ can only grow and Lemma 2.19 applies unchanged. This is the relation in which the branch witnesses w_2 supplied by $\mathcal{E}_{\text{DEC}}$ (Protocol 18) lives; cf. Remark 3.23.
-

Given $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$ and $V_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$, Lemma 2.19 ([FMN23, Lemma 7.1]) guarantees the existence of the extractor $E_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$. We define $E(\text{pp}, \mathbf{s}, u_1, \text{st}^*)$ as the algorithm that executes $E_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$ by simulating calls to $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$, and we construct an extractor $\mathcal{E}(\text{pp}, \mathbf{s}, u_1, \text{st})$ for the adversary $(\mathcal{A}, \mathcal{P}^*)$ to prove the first condition of a weak interactive reduction is satisfied. On input st , each iteration of $\mathcal{E}$ generates its own prefix: it resumes $\mathcal{P}^*$ from st with fresh randomness, samples an attempt index $\tau \xleftarrow{\$} [W_{\text{max}}]$, and lets $\mathcal{P}^*$ run the Rej_1 loop, taking $\text{st}^* := (\text{st}_{\mathcal{P}^*}, \text{tr}_{\text{pre}})$ to be the prover state and mask-commitment prefix of the τ -th attempt, and skipping to the next iteration if $\mathcal{P}^*$ halts or aborts before initiating that attempt; only then does it invoke $E(\text{pp}, \mathbf{s}, u_1, \text{st}^*)$. The prefix is thus drawn afresh in each of the N iterations, and is *not* the prefix of the first accepting attempt; see the discussion following Protocol 17.

Protocol 17 $\mathcal{E}(\text{pp}, \mathbf{s}, u_1, \text{st})$:

- 1: Set $N := \lambda^2 W_{\text{max}}$ (polynomial in λ).
- 2: **for** $\iota \in [N]$ **do**
- 3: Sample $\tau \xleftarrow{\$} [W_{\text{max}}]$. Resume $\mathcal{P}^*$ from st with fresh randomness and let it run the Rej_1 loop of Protocol 7; if $\mathcal{P}^*$ halts or aborts before initiating its τ -th attempt, **continue**. Otherwise let $\text{tr}_{\text{pre}} = (C_{\mathbf{y},0}, \{C_{\mathbf{y},j}\}_{j \in [t]})$ be the mask-commitment prefix of the τ -th attempt and $\text{st}_{\mathcal{P}^*}$ the corresponding prover state. Set $\text{st}^* := (\text{st}_{\mathcal{P}^*}, \text{tr}_{\text{pre}})$.
- 4: result $\leftarrow E(\text{pp}, \mathbf{s}, u_1, \text{st}^*)$.
- 5: **if** $u_1 = \perp$ or result $= \perp$ **then**
- 6: **continue**.

```

7:   end if
8:   Parse  $(\vec{c}, w'), (\vec{c}_1, w'_1), \dots, (\vec{c}_{K+k}, w'_{K+k}) \leftarrow \text{result}$ .
9:   Parse  $(z, \{s_{1,j}, s_{2,j}, y_j\}_{j \in [t]}) \leftarrow w'$  and  $\rho_1, \dots, \rho_{K+k} \leftarrow \vec{c}$ .
10:  for  $i \in [K+k]$  do
11:    Parse  $(z^{(i)}, \{s_{1,j}^{(i)}, s_{2,j}^{(i)}, y_j^{(i)}\}_{j \in [t]}) \leftarrow w'_i$  and  $\rho_1^{(i)}, \dots, \rho_{K+k}^{(i)} \leftarrow \vec{c}_i$ .
12:    Assign  $z_i \leftarrow (\rho_i - \rho_i^{(i)})^{-1} \cdot (z - z^{(i)})$ .
13:    Assign  $s_{1,j}^i \leftarrow (\rho_i - \rho_i^{(i)})^{-1} \cdot (s_{1,j} - s_{1,j}^{(i)})$ , for  $j \in [t]$ .
14:    Assign  $s_{2,j}^i \leftarrow (\rho_i - \rho_i^{(i)})^{-1} \cdot (s_{2,j} - s_{2,j}^{(i)})$ , for  $j \in [t]$ .
15:    Assign  $y_j^i \leftarrow (\rho_i - \rho_i^{(i)})^{-1} \cdot (y_j - y_j^{(i)})$ , for  $j \in [t]$ .
16:  end for
17:  Parse  $(c_i, x_i, r, \{\text{Commit}(y_j^i)\}_{j \in [t]})_{i \in [K+k]} \leftarrow u_1$ .
18:  return  $w_1 := (z_i, \{s_{1,j}^i, s_{2,j}^i, y_j^i\}_{j \in [t]})_{i \in [K+k]}$  if
   $(s; c_i, x_i, r, \{\text{Commit}(y_j^i)\}_{j \in [t]}; \{y_j^i\}_{j \in [t]}, \{s_{1,j}^i\}_{j \in [t]}, \{s_{2,j}^i\}_{j \in [t]}, z_i)_{i \in [K+k]} \in \text{CE}_{\text{com}}^{\text{rlx}, 3}(q/2, \mathcal{L}, \beta_{\text{rlx}}^{(3)}, \text{Commit})^{K+k}$ ,
19:  and continue otherwise.
20: end for
21: return  $\perp$ .

```

We highlight two main differences from [NS26, Appendix B.3], both forced by the Rej_1 loop. First, the attempt index τ is sampled uniformly from $[W_{\max}]$ rather than taken to be the first accepting attempt. Conditioning on acceptance biases the prefix distribution towards prefixes on which $\mathcal{P}^*$ is unusually cooperative, and no union bound over attempts would then be available. Second, the outer loop amplifies. The membership test in the penultimate step is decidable in polynomial time from $(\text{pp}, \mathbf{s}, u_1, w_1)$, consisting only of norm bounds and linear commitment identities, which is what makes amplification implementable.

Because st^* fixes $\text{tr}_{\text{pre}} = (C_{\mathbf{y},0}, \{C_{\mathbf{y},j}\}_{j \in [t]})$, the mask $\mathbf{y}$ and its salts and commitments are identical across every branch $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}(\vec{c})$ we rewind to, by construction. In particular, the opening $((c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}), \mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$ of $C_{\mathbf{y},0}$ produced at the end of each such branch is the same across branches, by binding of Commit (cf. Remark B.9). Both components matter and for different clauses of Definition 3.21: $c_{\mathbf{y}}$ cancels in $\mathcal{L}(\mathbf{z} - \mathbf{z}^{(i)})$ and $\mathbf{x}_{\mathbf{y}}$ in $\mathcal{L}_{\text{in}}(\mathbf{z} - \mathbf{z}^{(i)})$, so that the extracted $\mathbf{z}_i$ satisfies $c_i = \mathcal{L}(\mathbf{z}_i)$ and $\mathbf{x}_i = \mathcal{L}_{\text{in}}(\mathbf{z}_i)$. Had $\mathbf{x}_{\mathbf{y}}$ been left as an unbound post-challenge message, the second of these would fail with no extractable contradiction; cf. Remark B.3.

We already modified the adversary $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$, so we must relate the accepting probability of $V_{(\text{pp}, \mathbf{s}, u_1, \text{st}^*)}$ to $\epsilon(\mathcal{A}, \mathcal{P}^*)$. Lemma 3.20 is not available for this purpose as it describes the honest sampling procedure, whereas $\mathcal{P}^*$ need neither draw $\mathbf{y}$ from $D_{\mathbf{s}}^{n_{\mathbb{F}}}$ nor evaluate Rej_1 faithfully, and may decline to complete an attempt whenever the challenge block it has just received is unfavourable. Such issue motivates the introduction of the attempt budget $W_{\max}$. Fix $\theta := (\text{pp}, \mathbf{s}, u_1, \text{st})$ and write $p(\theta)$ for the conditional success probability of $\langle \mathcal{P}^*, \mathcal{V} \rangle$, so that $\epsilon(\mathcal{A}, \mathcal{P}^*) = \mathbb{E}_{\theta}[p(\theta)]$. For $\tau \in [W_{\max}]$ let Reach_{τ} be the event that $\mathcal{P}^*$ initiates a τ -th attempt, let st_{τ}^* be the corresponding prefix, and put

$$p_{\tau}(\text{st}_{\tau}^*) := \Pr_{\vec{c}}[V_{(\text{pp}, \mathbf{s}, u_1, \text{st}_{\tau}^*)}(\vec{c}, A_{(\text{pp}, \mathbf{s}, u_1, \text{st}_{\tau}^*)}(\vec{c})) = 1].$$

A successful execution completes acceptingly at some attempt, and by Protocol 7 there are at most $W_{\max}$ of them, so the union bound gives

$$p(\theta) \leq \sum_{\tau \in [W_{\max}]} \mathbb{E}[\mathbf{1}_{\text{Reach}_{\tau}} p_{\tau}(\text{st}_{\tau}^*)] = W_{\max} \cdot \mathbb{E}_{\tau \leftarrow [W_{\max}]}[\mathbf{1}_{\text{Reach}_{\tau}} p_{\tau}(\text{st}_{\tau}^*)].$$

For each fixed st_{τ}^* , Lemma 2.19 applies to $A_{(\text{pp}, \mathbf{s}, u_1, \text{st}_{\tau}^*)}$ with $\ell = K + k$ and produces the required $\ell + 1$ transcripts with probability at least $p_{\tau}(\text{st}_{\tau}^*) - (K + k)/|\mathcal{C}|$, using at most $\ell + 1$ queries in expectation. That bound is linear in $p_{\tau}(\text{st}_{\tau}^*)$ and we can average over τ and over the run directly. Writing $a(\theta)$ for the success probability of a single iteration ι of the outer loop of Protocol 17, the two formulas above give

$$a(\theta) \geq \mathbb{E}_{\tau} \left[\mathbf{1}_{\text{Reach}_{\tau}} \left(p_{\tau}(\text{st}_{\tau}^*) - \frac{K+k}{|\mathcal{C}|} \right) \right] \geq \frac{p(\theta)}{W_{\max}} - \frac{K+k}{|\mathcal{C}|} = \frac{u(\theta)}{W_{\max}}, \quad u(\theta) := p(\theta) - \frac{W_{\max}(K+k)}{|\mathcal{C}|}.$$

At this point we discuss why we needed to introduce N . Conditioned on θ , the N iterations of the outer loop use independent randomness and the membership test of the penultimate step is efficiently decidable,

so the extractor fails only if every iteration fails, an event of probability at most $(1 - a(\theta))^N$. If $u(\theta) \leq 0$ there is nothing to prove, the left-hand side of the display below being non-negative; assume therefore $u(\theta) > 0$. We set $N = \lambda^2 W_{\max}$, which is polynomial in λ by Definition 2.2. At the working level $\lambda = 120$ and with $W_{\max} \leq 2^8$ at every instantiation (namely $W_{\max} = 206$ for $M \approx 3$ and $W_{\max} = 122$ for $M \approx 2$, cf. Remark 4.3), this is $N \leq 2^{21.8}$. Since $a(\theta) \geq u(\theta)/W_{\max}$ we have $Na(\theta) \geq \lambda^2 u(\theta)$, whence

$$1 - (1 - a(\theta))^N \geq 1 - e^{-Na(\theta)} \geq 1 - e^{-\lambda^2 u(\theta)} \geq u(\theta) - e^{-\lambda}.$$

For the last inequality write $x := \lambda^2 u(\theta)$ and distinguish three cases. If $u(\theta) \leq \lambda^{-2}$ then $x \leq 1$, and $1 - e^{-x} \geq x/2 = \lambda^2 u(\theta)/2 \geq u(\theta)$ for $\lambda \geq 2$. If $\lambda^{-2} < u(\theta) \leq \lambda^{-1}$ then $x > 1$, so $1 - e^{-x} > 1 - e^{-1} > \lambda^{-1} \geq u(\theta)$. If $u(\theta) > \lambda^{-1}$ then $x > \lambda$, so $1 - e^{-x} > 1 - e^{-\lambda} \geq u(\theta) - e^{-\lambda}$. Only this last case consumes the additive slack, and it does so by $e^{-\lambda}$, which is negligible. Averaging over θ and using $\mathbb{E}_\theta[u(\theta)] = \epsilon(\mathcal{A}, \mathcal{P}^*) - W_{\max}(K+k)/|\mathcal{C}|$ yields extraction probability at least $\epsilon(\mathcal{A}, \mathcal{P}^*) - W_{\max}(K+k)/|\mathcal{C}| - \text{negl}(\lambda)$. By Definition 2.2, the quantities $W_{\max}$, K and k are polynomial in λ while $1/|\mathcal{C}| = \text{negl}(\lambda)$, so $W_{\max}(K+k)/|\mathcal{C}| = \text{negl}(\lambda)$ and Definition 2.6 is satisfied. Finally, N is fixed in advance and independent of the run, so linearity of expectation bounds the expected running time of $\mathcal{E}$ by N times that of a single iteration, each of which is EPT; $\mathcal{E}$ is therefore EPT. We stress that it is the *fixed* bound N that makes this argument available. An outer loop repeated until success would have expected cost $\approx 1/a(\theta)$, which no protocol parameter bounds (cf. Remark B.5).

We can then proceed as in [NS26, Appendix B.3], recalling that the ABDLOP commitment scheme is linear in both the salts and the message, and deduce that the extractor $\mathcal{E}(\text{pp}, \mathbf{s}, u_1, \text{st})$ is EPT, succeeding in extracting witnesses

$$(\mathbf{s}; c_i, x_i, r, \{\text{Commit}(y_j^i)\}_{j \in [t]}; \{y_j^i\}_{j \in [t]}; \{s_{1,j}^i\}_{j \in [t]}; \{s_{2,j}^i\}_{j \in [t]}; z_i)_{i \in [K+k]} \in \text{CE}_{\text{com}}^{\text{rlx},3}(q/2, \mathcal{L}, \beta_{\text{rlx}}^{(3)}, \text{Commit})^{K+k}$$

with probability at least $\epsilon(\mathcal{A}, \mathcal{P}^*) - W_{\max}(K+k)/|\mathcal{C}| - \text{negl}(\lambda)$.

It remains to check that the extracted salts satisfy the clause of Definition 3.24 at $e = 3$. Fix $i \in [K+k]$. The two branches w' and w'_i that Protocol 17 combines were both accepted by Protocol 16, hence lie in $\text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit})$, with certifiers $\bar{c}, \bar{c}_i \in \bar{\mathcal{C}} \setminus \{0\}$ respectively, so $\|\bar{c} \mathbf{s}_{l,j}\|_2 \leq \beta_{\text{rlx}}$ and $\|\bar{c}_i \mathbf{s}_{l,j}^{(i)}\|_2 \leq \beta_{\text{rlx}}$. Put $\Delta_i := \bar{c} \bar{c}_i (\rho_i - \rho_i^{(i)}) \in \bar{\mathcal{C}}^{-3}$, which is nonzero. Then $\Delta_i \mathbf{s}_{l,j}^i = \bar{c} \bar{c}_i (\mathbf{s}_{l,j} - \mathbf{s}_{l,j}^{(i)})$ and

$$\|\Delta_i \mathbf{s}_{l,j}^i\|_2 \leq \|\bar{c}_i (\bar{c} \mathbf{s}_{l,j})\|_2 + \|\bar{c} (\bar{c}_i \mathbf{s}_{l,j}^{(i)})\|_2 \leq T_1(\mathcal{C})(\|\bar{c} \mathbf{s}_{l,j}\|_2 + \|\bar{c}_i \mathbf{s}_{l,j}^{(i)}\|_2) \leq 2T_1(\mathcal{C})\beta_{\text{rlx}} = \beta_{\text{rlx}}^{(3)},$$

as required. The three factors of Δ_i are precisely the two $\bar{c}$'s of Lemma 3.6(2) and the one challenge difference of Lemma 2.19. The witness-norm clause of $\mathcal{R}'_2$ remains the vacuous $q/2$ inherited from [NS26, Appendix B.3], since coordinate-wise inversion certifies no bound on $\mathbf{z}_i$; that bound is supplied instead, within the same round, by the norm-check polynomial $\text{NC}(\bar{X})$ of Protocol 6, Step 3 (cf. Remark B.4.(1)).

For the second requirement of a weak interactive reduction, fix $\mathcal{A} = (\mathcal{B}, \mathcal{B}')$ as in Definition 2.6 and two instances u_1, u'_1 with $\phi(u_1) = \phi(u'_1)$, i.e. agreeing on $(c_i)_{i \in [K+k]}$. Since ψ retains only the $\mathbf{z}_i$, the claim to be proved is that the two extractions agree on those, and this is the argument of [NS26, Appendix B.3] verbatim: $c_i = \mathcal{L}(\mathbf{z}_i)$ together with the relaxed binding of $\mathcal{L}$ pins $\mathbf{z}_i$, up to $\epsilon(B, \mathcal{C})$.

We stress that the corresponding statement for the full witness is *false*, which is why ψ is needed. Neither remaining component is a function of $\phi(u_1)$. The hints satisfy $y_j^i = \bar{M}_j \mathbf{z}_i(r)$ and r is part of u_1 , not of $\phi(u_1)$, so two instances agreeing under ϕ but carrying distinct evaluation points yield distinct hints; and the salts extracted at Steps 1 and 17 are the ones $\mathcal{P}^*$ freshly sampled in that run, on which nothing in u_1 imposes a constraint beyond the commitment identity itself. An alternative repair would be to enlarge ϕ so as to determine them, but ϕ would then have to include r and the $C_{i,j}$, and clause (1) of Definition 2.7 would have to be re-established for Π'_{RLCS} against that larger ϕ , which fails for the same reason: r' is verifier randomness and the output commitments carry fresh salts, so two runs do not agree on them.

The knowledge-soundness error is

$$\varepsilon_{\text{RLC}} \leq \varepsilon_{\text{cw}} + \epsilon(B, \mathcal{C}) + \epsilon(\text{Commit}, \beta_{\text{rlx}}, \mathcal{C}) + \epsilon_{\text{bind}}(\tilde{B}) + ((K+k)t + t)\varepsilon_{\text{ABDLOP}}^{(1)},$$

the sum of the coordinate-wise extraction loss $W_{\max}(K+k)/|\mathcal{C}|$ of Lemma 2.19 inflated by the attempt budget, the $(2B, \mathcal{C})$ -relaxed-binding error of the Ajtai commitment scheme $\mathcal{L}$, the binding error of the ABDLOP commitment scheme Commit at the relaxed bound β_{rlx} (Definition 3.22) and the strong sampling set $\mathcal{C}$, the $\tilde{B}$ -binding error arising from the two-opening argument for $C_{y,0}$ (Remark B.9), and the knowledge-soundness error of the $(K+k)t + t$ ABDLOP proofs of knowledge of Steps 1 and 17, through which the

extractor $\mathcal{E}'$ above recovers the salts. Their sum is negligible in λ (cf. also Proposition 4.17), concluding the proof as in [NS26, Appendix B.3].

Remark B.2 (Why the loop is capped). *The bound $W_{\max}$ of Protocol 7 is necessary for proving soundness. Without it, an EPT $\mathcal{P}^*$ admits only a bound $\mathbb{E}[W] \leq \bar{W}$ on its expected number of attempts, so truncating the analysis at a budget $p_{\lambda,W} := p(\lambda)\bar{W}$ leaves a residue $\Pr[W > p_{\lambda,W}] \leq 1/p(\lambda)$ by Markov's inequality, which is an inverse polynomial, not a negligible function. With the cap, $\Pr[W > W_{\max}] = 0$ by construction, and Definitions 2.6 and 2.7 are used exactly as in [NS26, Definitions 16–17], with Proposition 2.8 applied (almost) verbatim.*

The price is a completeness error of $2^{-\lambda}$ per capped loop. The same argument, and the same price, apply verbatim to the retry budget $W_{\max}^{\text{PoK}}$ (Equation (B.1)) of the ABDLOP proofs of knowledge, which we assume to be enforced through each proof of knowledge. There too an uncapped loop would leave an inverse-polynomial Markov residue in the union bound over which attempt a malicious prover elects to complete. Since our protocols cap $1 + N_{\text{PoK}}$ loops in all, the total is the $\varepsilon_c^{\text{LBF}}$ of Proposition 4.15.

We stress that these caps bound the number of attempts an honest prover makes and the number of first messages a malicious one may select among. However, they do not bound, and must not be used to bound, the number of times an extractor rewinds. That count is governed by the adversary's success probability, and the rewinding loops of Protocol 18 are accordingly left uncapped and analysed in expectation; cf. Remark B.5.

Remark B.3 ($\mathbf{x}_y$ must be bound before $\bar{\rho}$, not merely sent after it). *We cannot leave $\mathbf{x}_y = \mathcal{L}_{\text{in}}(\mathbf{y})$ as a plain (post-challenge) message even though it is public data that the verifier folds into $\mathbf{x}$ and never checks a norm on. The problem arises as it would break the extraction argument of Lemma 4.10.*

The extractor rewinds to two branches differing only in coordinate i and forms $\mathbf{z}_i := (\rho_i - \rho_i^{(i)})^{-1}(\mathbf{z} - \mathbf{z}^{(i)})$. For the commitment clause of Definition 3.21 this is enough: c_y is fixed across branches by binding of Commit, so it cancels in the difference and $\mathcal{L}(\mathbf{z} - \mathbf{z}^{(i)}) = (\rho_i - \rho_i^{(i)})c_i$. For the instance clause $\mathbf{x}_i = \mathcal{L}_{\text{in}}(\mathbf{z}_i)$ it is not, since

$$\mathcal{L}_{\text{in}}(\mathbf{z} - \mathbf{z}^{(i)}) = (\rho_i - \rho_i^{(i)})\mathbf{x}_i + (\mathbf{x}_y - \mathbf{x}_y^{(i)}),$$

and an unbound $\mathbf{x}_y$ may be chosen per branch, indeed defined as $\mathcal{L}_{\text{in}}(\mathbf{z}) - \sum_j \rho_j \mathbf{x}_j$ after the fact. Two branches that disagree there do not yield a contradiction to extract from. Both $\mathbf{z}$ and $\mathbf{z}^{(i)}$ are then short preimages of different targets, so no MSIS solution is produced and the extractor simply fails.

Committing to the pair $(c_y, \mathbf{x}_y)$ at Step 6 removes the issue at once. Binding of Commit fixes $\mathbf{x}_y$ before $\bar{\rho}$ is drawn, exactly as it already fixes c_y , and the displayed difference collapses to $(\rho_i - \rho_i^{(i)})\mathbf{x}_i$. The cost is $n_{R,\text{in}}$ further BDLOP message slots on the one commitment $C_{y,0}$, i.e. $n_{R,\text{in}} \log_2 q$ bits on its $\mathbf{t}_B$ component; the values themselves are free, $\mathbf{x}_y$ being already sent in the clear at Step 12. It would be wrong, however, to read this as leaving the transmitted opening untouched. The message $(c_y, \mathbf{x}_y)$ has length $\kappa + n_{R,\text{in}}$, which overflows the ℓ message slots of a single commitment, so $C_{y,0}$ is in fact a tuple of $\tilde{N} = \lceil (\kappa + n_{R,\text{in}})/\ell \rceil$ commitments, each carrying its own salt pair into the revealed opening; see Remark 4.21.

Similarly, sending $\mathbf{x}_y$ in the clear before $\bar{\rho}$ is not an acceptable alternative. In a rejected attempt the conditional law of $\mathbf{y}$ depends on $\mathbf{v}$, and hence on the witness, which is precisely the leak Remark B.9 exists to prevent; the same objection that forces c_y under a commitment forces $\mathbf{x}_y$ under one too.

The analogous question for $\{C_{y,j}\}_{j \in [t]}$ does not arise. There the homomorphic check of Step 21, together with binding of Commit, already pins $y_{y,j}$ before $\bar{\rho}$, and the hint clause of Definition 3.21 then follows by $R_{\mathbb{F}}$ -linearity in the difference exactly as the commitment clause does.

B.3. Proof of Lemma 4.12. Before proving Lemma 4.12, let us single out two structural considerations.

Remark B.4.

- (1) Π'_{DEC} performs no independent check that each extracted piece z_i satisfies $\|z_i\|_{\infty} < b$. This bound is guaranteed because split_b is applied to an already norm-bounded z by an honest prover, and is otherwise enforced downstream by the norm-check polynomial $\text{NC}(\vec{X})$ of Π'_{R1CS} 's Sum-Check (Protocol 6, Step 3). Π'_{DEC} is therefore sound only as a component of the specific composition $\Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$ analyzed in Theorem 4.13, and should not be treated as an independently reusable building block without re-deriving norm-bound enforcement for its new context.
- (2) Π'_{DEC} is not deterministic. Step 5 carries a verifier challenge and the extractor rewinds it.

We can now prove the lemma.

Completeness: The proof in [NS26, Appendix B.4] applies mutatis mutandis to Steps 1–4. Step 5 is complete by Lemma 3.6, up to the rejection-sampling aborts of Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$; capping the number of attempts at $W_{\text{max}}^{\text{PoK}}$ as in Equation B.1 makes completeness statistical with error $2^{-\lambda}$, matching the error already incurred by Protocol 7.

Public Coin: the verifier's only randomness is that of Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ at Step 5, which is public coin.

Knowledge Soundness: Let $(\mathcal{A}, \mathcal{P}^*)$ be an EPT adversary with $\epsilon(\mathcal{A}, \mathcal{P}^*) \geq 1/\text{poly}(\lambda)$. The extractor is Protocol 18; we highlight in green all changes from [NS26, Appendix B.4].

Protocol 18 $\mathcal{E}(\text{pp}, \mathbf{s}, u_1 := (c, x, r, \{C_j\}_{j \in [t]}), \text{st})$:

- 1: Execute encoder $(\text{pk}, \text{vk}) \leftarrow \mathcal{K}(\text{pp}, \mathbf{s})$.
 - 2: Simulate $(u_2, w_2) \leftarrow \langle \mathcal{P}^*(\text{pk}, u_1, \text{st}), \mathcal{V}(\text{vk}, u_1) \rangle$.
 - 3: **if** $u_2 = \perp$ **then**
 - 4: **return** $\perp$.
 - 5: **end if**
 - 6: Parse $(\{y_{i,j}\}_{j \in [t]}, \{s_{l,i,j}\}_{(l,j) \in [2] \times [t]}, \{z_i\}_{i \in [k]}) \leftarrow w_2$.
 - 7: Run the knowledge extractor $\mathcal{E}_{\text{ABDLOP}}$ of Lemma 3.6 (cf. [LNP22, Proof of Lemma 3.1]) on the $\Pi_{\text{many}}^{(1)}$ instance of Step 5 of Protocol 8, with rewindable black-box access to $\mathcal{P}^*$ and with (u_1, st) (hence u_2) held fixed. It rewinds $\mathcal{P}^*$ on the challenge of that step alone and, except with probability $\epsilon_{\text{ABDLOP}}^{(1)}$, returns two accepting transcripts sharing their first message $(\mathbf{w}, \mathbf{v})$ and differing in the challenge. If it aborts, abort as well and return $\perp$.
 - 8: Apply Lemma 3.6 to the two transcripts to obtain $(\bar{\mathbf{s}}_{1,j}, \bar{\mathbf{s}}_{2,j}, \bar{c})$ with $\|\bar{c}\bar{\mathbf{s}}_{l,j}\|_2 \leq \beta_{\text{rlx}}$ and $C_j = \text{Commit}(\bar{y}_j, \bar{\mathbf{s}}_j)$ for $\bar{y}_j := \mathbf{t}_{B,j} - \mathbf{B}\bar{\mathbf{s}}_{2,j}$; by item (3) of that lemma, $\bar{y}_j = \sum_{i \in [k]} b^{i-1} y_{i,j}$.
 - 9: **return** $w_1 := (\{\sum_i b^{i-1} y_{i,j}\}_{j \in [t]}, \{\bar{\mathbf{s}}_{l,j}\}_{(l,j) \in [2] \times [t]}, \sum_i b^{i-1} z_i)$.
-

Three points complete the argument:

(i) $\mathcal{E}$ is EPT: We condition on the first execution being accepting with first message $(\mathbf{w}, \mathbf{v})$, and write $\epsilon_{(\mathbf{w}, \mathbf{v})} := \Pr_c[\mathcal{V} \text{ accepts} \mid (\mathbf{w}, \mathbf{v})]$. The number of rewinds is then geometric with parameter $\epsilon_{(\mathbf{w}, \mathbf{v})}$, of expectation $\epsilon_{(\mathbf{w}, \mathbf{v})}^{-1}$, while the branch is reached at all only with probability $\epsilon_{(\mathbf{w}, \mathbf{v})}$. Averaging over first messages gives an expected number of rewinds at most 1, *uniformly* in $\epsilon(\mathcal{A}, \mathcal{P}^*)$. This is the standard rewinding argument underlying Lemma 3.6, which we invoke rather than re-derive. The bound is uniform in $\epsilon(\mathcal{A}, \mathcal{P}^*)$, which is what the composition needs; see Remark B.5. The failure probability of this step is $\epsilon_{\text{ABDLOP}}^{(1)}$, which already carries the factor $W_{\text{max}}^{\text{PoK}}$ accounting for the at most $W_{\text{max}}^{\text{PoK}}$ first messages a malicious prover may select among before completing an attempt (§4.3.2).

(ii) w_2 is the same across the rewind branches: since u_2 is fixed at Step 3 of Protocol 8 and binding of $\mathcal{L}$ and Commit pins the openings; this costs the term $\epsilon(\text{Commit}, \bar{B}, C)$.

(iii) $(\mathbf{s}; u_1; w_1) \in \text{CE}_{\text{com}}^{\text{rlx}}(B, \mathcal{L}, \beta_{\text{rlx}}, \text{Commit})$: the recompositions are correct by Lemma 3.6(3), $\|\sum_i b^{i-1} z_i\|_\infty < B$ by $\mathcal{R}_2$ -membership of w_2 , and the salt clause is the relaxed one supplied by Lemma 3.6(2). This is the relation in which Protocol 16 accepts, so the branch witnesses handed to Lemma 2.19 are of the right type; cf. Remark 3.23 and, for the scope of the relaxation, Remark 4.8.

Remark B.5. For the sake of clarity, let us explain why the extractor does not need to be capped. $W_{\text{max}}^{\text{PoK}}$ of Equation B.1 is the honest prover's rejection-sampling retry cap inside an ABDLOP proof of knowledge. It is public, enforced by $\mathcal{V}$, and its purpose is to make budget exhaustion a witness-independent event of probability at most $2^{-\lambda}$, charged to completeness (Remark 4.3, Proposition 4.15). The number of rewinds an extractor needs is a different quantity altogether. it is governed by $\epsilon(\mathcal{A}, \mathcal{P}^*)$, which no protocol parameter bounds. Capping the rewinding at $W_{\text{max}}^{\text{PoK}}$ would establish knowledge soundness only against provers of success probability $\gtrsim 1/W_{\text{max}}^{\text{PoK}}$, i.e. it would establish nothing.

Note that the cap is not needed to compose. In the knowledge-soundness proof extraction runs backwards, so the adversary $\mathcal{A}$ that Lemma 2.19 rewinds on the folding coefficients $\bar{c}$ is the one that runs $\mathcal{P}^*$ through Protocol 8 and applies $\mathcal{E}_{\text{DEC}}$ to the result (this is how it produces the branch witness w_2 whose $\text{CE}_{\text{com}}^{\text{rlx}}$ -membership Protocol 16 checks). That lemma asks only that $\mathcal{A}$ be EPT, which point (i) of the proof of Lemma 4.12 in §B.3 supplies. The two randomnesses are independent (the coins $\mathcal{E}_{\text{DEC}}$ spends rewinding the Step 5 challenge

have nothing to do with $\vec{c}$) so the expected total work is the product of the two expectations (Wald's identity), both polynomial.

The cap does enter the knowledge-soundness accounting. A malicious prover may choose which of its at most $W_{\max}^{\text{PoK}}$ attempts to complete, and so which first message the extractor is handed, inflating its conditional success probability by that factor. That inflation is already inside $\varepsilon_{\text{ABDLOP}}^{(1)}$ (§4.3.2) and is the exact analogue of the role $W_{\max}$ plays, through the uniform choice of the attempt index τ , in Protocol 17. What Remark B.2 establishes is that the mask loop of Protocol 7 must be capped so that this union bound is over a bounded number of attempts.

B.4. Proof of Theorem 4.13. The composition $\Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{RCS}}$ is a reduction of knowledge by Lemma 4.11 and Lemma 4.12; the result follows for Π_{LBF} . The completeness error is the sum of the three, by the composition clause of Proposition 2.8, and is computed in Proposition 4.15. Note that Protocols 12 adds nothing to it, their only extra step being the local sampling of Protocol 11, which is rejection-free in the relevant sense (its restart condition $\|\mathbf{z}\|_{\infty} > \tau_{\lambda, \xi} 5$ fails only with the negligible probability of Lemma 3.19 and is a local resampling, not an interaction).

Assume now that $K = 1$. We prove that Π_{LBF} satisfies Definition 2.11 in the setting $l = 1$, $\mathcal{R}' = \text{CCS}(b, \mathcal{L})$ (Definition 2.21), and $\mathcal{R} = \mathcal{R}_2 = \text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ (Definitions 3.21), and then the same statement for Π_{LBF} over the unpadding $\mathcal{R}' = \text{CCS}^{\circ}(b)$ of Definition 3.28, which is the reading that gives zero-knowledge. For this purpose we construct a simulator $\mathcal{S}$ which simulates the transcript and output witness for the interactive reduction $\Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{RCS}}$ given a tuple of instances and public parameters.

Remark B.6 (Naming). Throughout this appendix $u_1 = (c_1, \mathbf{x}_1)$ denotes the $\mathcal{R}' = \text{CCS}$ instance, which in the indexing of Definition 2.11 is u_2 ; since $l = 1$ the $\mathcal{R}^{l-1}$ component of that definition is empty, so no ambiguity arises. The k pairs the prover samples are indexed $i \in [K + 1, K + k]$ and written with the superscript blind.

The first six hybrids below establish Definition 2.11 over the committed relation and are insensitive to how the witness was produced. Only the last one, H_6 , needs the unpadding reading; we flag this explicitly where it occurs.

Before giving the simulator we record precisely which quantities $\mathcal{S}$ must be consistent with (recall Definition 2.21):

- with $l = 1$, Definition 2.11 hands $\mathcal{S}$ only the public $\text{CCS}(b, \mathcal{L})^K$ instance $u_1 = (c_1, \mathbf{x}_1)$, $K = 1$, with

$$c_1 = \mathcal{L}(\mathbf{z}_1), \quad \mathbf{x}_1 = \mathcal{L}_{\text{in}}(\mathbf{z}_1), \quad \|\mathbf{z}_1\|_{\infty} < b,$$

for a witness $\mathbf{z}_1$ that $\mathcal{S}$ does not receive;

- every other public value occurring in the composed protocol is produced *inside* the execution and is therefore free for $\mathcal{S}$ to choose: the blinding instances $(c_i^{\text{blind}}, \mathbf{x}_i^{\text{blind}}, \{C_{i,j}^{\text{blind}}\})$, the Π'_{RCS} output commitments $\{\text{Commit}(y'_{i,j}, \mathbf{s}'_{i,j})\}_{i \in [K+k]}$ (including $i = 1$, whose salts the honest prover samples freshly at Step 9 of Protocol 6), the mask block $(c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}, \{C_{\mathbf{y},j}\})$, and all Π'_{DEC} data.

Remark B.7.

- (1) Some public value must be fixed after $\vec{\rho}$ is known. More precisely, the verifier computes the output instance component as $c' = \rho_1 c_1 + \sum_{i \in [k]} \rho_{K+i} c_i^{\text{blind}} + c_{\mathbf{y}}$, while checking membership of the output pair in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k$ requires $c' = \mathcal{L}(\mathbf{z}')$ for the bounded witness $\mathbf{z}'$ that Definition 2.11 also outputs. A simulator that fixed every public value before $\vec{\rho}$ and then produced a short $\mathbf{z}'$ satisfying this identity would, in particular, have to find a short preimage of $\rho_1 c_1$ under $\mathcal{L}$, contradicting the relaxed binding of the Ajtai commitment. Late binding of one public value is therefore unavoidable. The witness components and the public inputs $\mathbf{x}_i^{\text{blind}}$ carry a visible norm structure ($\|\mathbf{x}_i^{\text{blind}}\|_{\infty} < b$, i.e. ternary entries for $b = 2$), so solving either of them backwards from a linear equation yields a value that would not fit the correct distribution and be rejected on sight. However, we still have some degree of freedom. By contrast $c_{\mathbf{y}} \in \mathbb{C}$ is a compact Ajtai commitment which is never opened, never norm-checked, and enters only $\mathcal{V}$'s linear combination. It is the unique slack variable in the protocol that carries no norm obligation.
- (2) The verifier $\mathcal{V}^*$ is honest-but-curious and public-coin, meaning that every challenge it sends is a deterministic function of its coin string ω alone, and only its final output st may depend on the

transcript. Since $\mathcal{S}$ samples $\omega' \xleftarrow{\$} \Omega$ itself, it may derive all challenges before producing any prover message.

Since the simulator includes rejection sampling, we denote by τ the repetition index.

Protocol 19 Forward simulator for $\Pi'_{\text{DEC}} \circ \Pi'_{\text{RLC}} \circ \Pi'_{\text{R1CS}}$: $\mathcal{S}$

Input: pp , structure $\mathbf{s}$, and the public $\text{CCS}(b, \mathcal{L})^K$ instance $u_1 = (c_1, \mathbf{x}_1)$, $K = 1$.

▷ Naming: r' is the point produced by Π'_{R1CS} ; Protocols 7 and 8 receive it as their input point and call it r . We follow each protocol's own name, so $r = r'$ throughout.

Output: $(\text{st}, \mathbf{w}')$.

- 1: Sample $\omega' \xleftarrow{\$} \Omega$ and derive from it all challenges of $\mathcal{V}^*$: $\alpha, \gamma, (\delta_0, \delta_1), \zeta, r', \{\mu_\ell\}_{\ell \geq 2}$ and the challenge blocks $\tilde{\rho}^{(\tau)} \in \mathcal{C}^{K+k}$, $\tau = 1, 2, \dots$ (Remark B.7).
- 2: **Blinding pairs (honest, bounded).** $(u_i^{\text{blind}}, w_i^{\text{blind}})_{i=K+1}^{K+k} \leftarrow \text{sample}_{\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit})^k}^{\text{blind}}(\text{pp}, \mathbf{s})$ (Protocol 11), using *exactly* the sampler the honest prover uses at Step 1 of Protocol 12. In particular $\|z_i^{\text{blind}}\|_\infty < b$, $\|s_{l,i,j}^{\text{blind}}\|_\infty < \tilde{B}$ and $c_i^{\text{blind}} = \mathcal{L}(z_i^{\text{blind}})$, $\mathbf{x}_i^{\text{blind}} = \mathcal{L}_{\text{in}}(z_i^{\text{blind}})$.
- 3: **Fake R1CS witness.** Sample $\hat{\mathbf{w}}_1 \xleftarrow{\$} \{\mathbf{w} : \|\mathbf{w}\|_\infty < b\}$ and set $\hat{\mathbf{z}}_1 := [\mathbf{x}_1, \hat{\mathbf{w}}_1]$. ▷ Same norm bound b as an honest witness, and $\mathcal{L}_{\text{in}}(\hat{\mathbf{z}}_1) = \mathbf{x}_1$ by construction. We do *not* have $\mathcal{L}(\hat{\mathbf{z}}_1) = c_1$, and never use it; $\hat{\mathbf{z}}_1$ is never claimed to be an opening of c_1 nor to satisfy the R1CS structure.
- 4: **Simulated Π'_{R1CS} transcript.** For all $i \in [K+k]$, $j \in [t]$ put $\hat{\mathbf{z}}_i := z_i^{\text{blind}}$ for $i > K$, compute $\hat{y}'_{i,j} := \tilde{M}_j \hat{\mathbf{z}}_i(r')$, sample fresh salts $\hat{s}'_{i,j}$ with $\|\hat{s}'_{i,j}\|_\infty < \tilde{B}$ and set $C'_{i,j} := \text{Commit}(\hat{y}'_{i,j}, \hat{s}'_{i,j})$ *honestly*.
- 5: Produce every remaining message of Π'_{R1CS} the ABDLOP proofs of knowledge (linear, quadratic and anchoring), the degree-0 commitments $\hat{c}_{i,j}$, the claimed sum σ , the coeffs commitment, the Sum-Check round polynomials and the final value h) with the corresponding sub-simulators: ▷ $\mathcal{S}$ knows valid openings of all commitments it sends, only the statements proved about them are false, so the sub-proofs come from HVZK simulators.

Run the honest prover of Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$ for the commitments $\{C_{i,j}\}_{j \in [t]}$, $i \in [K+1, K+k]$. ▷ These belong to the blinding pairs $\mathcal{S}$ generated itself at Step 2, so $\mathcal{S}$ knows $(y_{i,j}, s_{i,j})$ and the resulting transcript is identically distributed.

Simulate, via $\mathcal{S}_{\text{ABDLOP}}$, the t invocations of Step 1 of Protocol 7 on $\{C'_{1,j}\}_{j \in [t]}$. ▷ Unlike the blinding pairs above, these cannot be run honestly: $\mathcal{S}$ holds the opening $(\hat{y}'_{1,j}, \hat{s}'_{1,j})$ of the *fake* witness $\hat{\mathbf{z}}_1$, not the real prover's, so an honest run would not be identically distributed. The same applies to the t invocations of Step 17 on $\{C_{y,j}\}$, simulated at the folding step below. These $2t$ are the only proofs of knowledge outside Π'_{R1CS} that $\mathcal{S}$ must simulate rather than execute.

Compute the sum T exactly as in Protocol 6, using only the carried-over evaluation hints $\{y'_{i,j}\}_{i \in \{K+1, \dots, K+k\}}$ fixed at Step 2 (simulator-sampled). ▷ We do *not* compute $F(\alpha) + \delta_0 \cdot \text{NC}(\alpha)$, unlike the honest prover. Sample the Sum-Check transcript directly: draw the round polynomials uniformly subject to $h_i(0) + h_i(1) = h_{i-1}(r'_{i-1})$ at degree profile $(d_i)_{i \in [\log_2 m]}$, anchored at the public target sum $\zeta P + T$, and read off the final-round value h (matching the notation of Protocol 5 and of Protocol 6, Step 8). ▷

By §4.1.1.1 this is *exactly* the real law and not merely close to it, so the step costs nothing. The map from the mask's coefficients to (claimed sum, non-constant round coefficients) is a bijection for odd q and $\zeta \in \mathbb{K}^\times$. No evaluation of Q at a real point is required. Since h is a public Sum-Check message rather than a committed value (cf. Remark 4.2.(8)), producing it is part of sampling the transcript itself. We do not invoke [XZZ⁺19, Theorem 3], whose simulator is statistical and assumes the mask is opened at r' .

Simulate the commitment to coeffs, sample $\sigma \leftarrow \mathbb{K}$ uniformly, and simulate the anchoring ABDLOP ZKPoK binding σ to it, via $\mathcal{S}_{\text{ABDLOP}}$ (resp. the simulator of Lemma 3.16).

For each $i \in [K+k]$, $j \in [t]$, sample a fresh salt and commit to a degree-0 ring element $\hat{y}_{i,j}$ freely chosen with $\text{ct}(\hat{y}_{i,j})$ arbitrary, via $\hat{c}_{i,j} := \text{Commit}(\hat{y}_{i,j})$. ▷ Mirrors Protocol 6's degree-0 step; cf. Lemma 3.13. Simulate, via $\mathcal{S}_{\text{ABDLOP}}$, the batched $\mathbb{Z}_q$ -linear ABDLOP proof of knowledge for $\{\text{ct}(\pi_\mu \cdot \hat{y}_{i,j}) = 0, \text{ct}(\hat{y}'_{i,j} - \hat{y}_{i,j}) = 0\}_{i \in [K+k], j \in [t]}$.

We compute the commitments $\{\hat{c}'_{i,sq}\}_{i \in [K+k]}$, $\{\hat{c}'_{i,\text{cube}}\}_{i \in [K+k]}$, and $\{\hat{c}'_{i,\text{prod}}\}_{i \in [K]}$ generating random salts. ▷ Built from the freely-chosen $\hat{y}_{i,1}, \hat{y}_{i,2}, \hat{y}_{i,3}$ of the previous step, not from any real square/cube/product of $y'_{i,1}, y'_{i,2}, y'_{i,3}$.

Simulate, via $\mathcal{S}_{\text{ABDLOP}}$ and the quadratic-relations simulator of Protocol $\Pi_{\text{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \dots, f_N))$, the proofs of knowledge for the batch of quadratic relations on $\hat{y}_{i,1}, \hat{y}_{i,2}, \hat{y}_{i,3}, \hat{c}'_{i,sg}, \hat{c}'_{i,cube}, \hat{c}'_{i,prod}$ (Step 15 of Protocol 6), and finally simulate the anchoring call Π^{anc} for $u^* := h - \text{eq}(r', \alpha) \cdot (c_F + \delta_0 \cdot c_N) - \delta_1 \cdot c_E$, where c_F, c_N, c_E are derived, exactly as $\mathcal{V}$ would, from the already-simulated $\hat{y}_{i,j}$'s and quadratic commitments, and h is the value simulated above. $\triangleright u^*$ is a public value $\mathcal{S}$ computes itself, exactly as the real $\mathcal{V}$ would; per the weight table of §3.2.1.1, the anchoring call certifies $u^* = \zeta \cdot S_{2,p}$ against the single salt block $\mathbf{s}_{P,2}$ underlying the already-simulated coeffs. Note that we do not consider any honest execution against $\mathcal{V}^*$ here. Every sub-protocol invoked is individually HVZK-simulatable, and we define $\mathcal{S}$ as the composition of their simulators rather than recomputing real openings.

6: **Forward Π'_{RLC} (capped rejection loop).**

7: **for $\tau = 1$ to W_{max} do**

(a) Read $\bar{\rho}^{(\tau)}$ from ω' and set $\mathbf{v}^{(\tau)} := \rho_1^{(\tau)} \hat{\mathbf{z}}_1 + \sum_{i \in [k]} \rho_{K+i}^{(\tau)} \mathbf{z}_i^{\text{blind}}$. $\triangleright \mathcal{V}^*$ is public coin

and, by Definition 2.11, every challenge is a deterministic function of the coin string ω' alone (not of the transcript). $\mathcal{S}$ sampled ω' itself (Remark B.7), hence knows $\bar{\rho}^{(\tau)}$ before producing any prover message of this attempt. This is the only place where that is used, and it is precisely the asymmetry isolated in Remark B.8.

(b) Sample $\mathbf{y}^{(\tau)} \leftarrow D_{\mathbf{s}}^{n_F}$ and set $\mathbf{z}^{(\tau)} := \mathbf{v}^{(\tau)} + \mathbf{y}^{(\tau)}$.

(c) Evaluate $\text{Rej}_1(\mathbf{z}^{(\tau)}, \mathbf{v}^{(\tau)}, \mathbf{s})$. $\triangleright \|\hat{\mathbf{z}}_1\|_{\infty} < b$ by Step 3, so $\mathbf{v}^{(\tau)} \in V$ and Lemma 3.20 applies. The acceptance probability satisfies $p(\mathbf{v}) \in [(1 - 2^{-\lambda})/M, 1/M]$ for every $\mathbf{v} \in V$.

(d) **If it rejects:** sample fresh salt blocks and send $C_{\mathbf{y},0}^{(\tau)} := \text{Commit}(0, \cdot)$ and $C_{\mathbf{y},j}^{(\tau)} := \text{Commit}(0, \cdot)$, $j \in [t]$. These are never opened. Continue.

(e) **If it accepts:** write $\tilde{\tau} := \tau$, drop the superscript, and set

$$c_{\mathbf{y}} := \mathcal{L}(\mathbf{z}) - \rho_1 c_1 - \sum_{i \in [k]} \rho_{K+i} c_i^{\text{blind}}.$$

Set likewise $\mathbf{x}_{\mathbf{y}} := \mathcal{L}_{\text{in}}(\mathbf{z}) - \rho_1 \mathbf{x}_1 - \sum_{i \in [k]} \rho_{K+i} \mathbf{x}_i^{\text{blind}}$. Sample a fresh salt block $(\mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$ with $\|\mathbf{s}_{1,\mathbf{y},0}\|_{\infty} < \tilde{B}$ and send the *honestly formed* commitment $C_{\mathbf{y},0} := \text{Commit}((c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}), \mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$. For $j \in [t]$ set $y_{\mathbf{y},j} := \widetilde{M_j \mathbf{y}^{(\tau)}}(r)$ and send $C_{\mathbf{y},j} := \text{Commit}(y_{\mathbf{y},j}, \cdot)$ under fresh salts. Exit the loop. $\triangleright c_{\mathbf{y}}$ is computed before $C_{\mathbf{y},0}$ is formed, and $C_{\mathbf{y},0}$ is opened to the value it was created with.

8: **end for**

9: **Budget exhaustion.** If no attempt was accepted, output the transcript truncated at the W_{max} -th challenge together with $\mathcal{V}$'s $\perp$, and halt. $\triangleright$ This mirrors Step 8 of Protocol 7, where $\mathcal{V}$ halts on $w > W_{\text{max}}$. The halt is part of $\mathcal{V}^*$'s view and must therefore be reproduced, not merely declared a simulator failure.

10: **Opening the mask block.** Send the opening $((c_{\mathbf{y}}, \mathbf{x}_{\mathbf{y}}), \mathbf{s}_{1,\mathbf{y},0}, \mathbf{s}_{2,\mathbf{y},0})$ of $C_{\mathbf{y},0}$, as at Step 12 of Protocol 7. $\mathcal{V}$'s recomputation and its norm check on the revealed salts pass by construction. $\triangleright$ In an *honest* execution $c_{\mathbf{y}}$ satisfies the very same identity, since $c_{\mathbf{y}} = \mathcal{L}(\mathbf{y})$, $\mathbf{z} = \mathbf{v} + \mathbf{y}$, $c_1 = \mathcal{L}(\mathbf{z}_1)$ and $c_i^{\text{blind}} = \mathcal{L}(\mathbf{z}_i^{\text{blind}})$. So $c_{\mathbf{y}}$ is the *same deterministic function* of $(\bar{\rho}, \mathbf{z}, c_1, \{c_i^{\text{blind}}\})$ in the real and the simulated world, and no property of $\mathcal{L}$ beyond $\mathbb{F}$ -linearity is invoked. At the level of instances $\mathcal{L}_{\text{in}}(\mathbf{z}) = \rho_1 \mathbf{x}_1 + \sum_i \rho_{K+i} \mathbf{x}_i^{\text{blind}} + \mathbf{x}_{\mathbf{y}}$ holds automatically, because $\mathcal{L}_{\text{in}}(\hat{\mathbf{z}}_1) = \mathbf{x}_1$.

11: **Folding, computed forward.** For $j \in [t]$ set $y_j := \sum_{i \in [K+k]} \rho_i \hat{y}'_{i,j} + y_{\mathbf{y},j}$ and $\mathbf{s}_{l,j} := \sum_{i \in [K+k]} \rho_i \hat{\mathbf{s}}'_{l,i,j} + \mathbf{s}_{l,\mathbf{y},j}$, and send $\text{Commit}(y_j, \mathbf{s}_j)$; by linearity of Commit this equals $\sum_i \rho_i C'_{i,j} + C_{\mathbf{y},j}$, so $\mathcal{V}$'s check of Step 21 of Protocol 7 passes identically.

Simulate via $\mathcal{S}_{\text{ABDLOP}}$ the t invocations of Step 17 of Protocol 7 on $\{C_{\mathbf{y},j}\}_{j \in [t]}$. $\triangleright$ They are simulated rather than executed for the reason given at the Step 1 invocations above. The message inside $C_{\mathbf{y},j}$ is altered by hybrid H_4 of the analysis.

12: **Forward Π'_{DEC} .** $(\mathbf{z}'_1, \dots, \mathbf{z}'_k) \leftarrow \text{split}_b(\mathbf{z})$, $\mathbf{s}'_{l,i,j}$ freshly sampled with $\|\mathbf{s}'_{l,i,j}\|_{\infty} < \tilde{B}$, $c'_i := \mathcal{L}(\mathbf{z}'_i)$, $\mathbf{x}'_i$ as $\mathcal{V}$ computes them, $y'_{i,j} := \widetilde{M_j \mathbf{z}'_i}(r)$, $C'_{i,j} := \text{Commit}(y'_{i,j}, \mathbf{s}'_{l,i,j})$. The check of Step 4 of Protocol 8 holds by construction, since $\sum_i b^{i-1} c'_i = \mathcal{L}(\mathbf{z}) = c'$ agrees with the value derived in the above Steps 6–8, and the proof of knowledge of Step 5 above is run *honestly*. $\triangleright \mathbf{z}$ is the value $\mathcal{S}$ itself formed, $\text{split}_b(\mathbf{z})$ is computed honestly, the salts are fresh and known, and the certified relation $y_j - \sum_i b^{i-1} y_{i,j} = 0$ holds genuinely, so the transcript is identically distributed and this invocation costs nothing.

13: Run $\mathcal{V}^*(\cdot; \omega')$ on the resulting transcript, let $\mathbf{st}$ be its output, and return $(\mathbf{st}, \mathbf{w}')$ with $\mathbf{w}' = (\mathbf{z}'_i, \{y'_{i,j}\}_j, \{\mathbf{s}'_{i,j}\}_j)_{i \in [k]}$.

Now we prove, in the order, that:

- (1) The output of $\mathcal{S}(\mathbf{pp}, \mathbf{s}, u_1)$ in Protocol 19 is computationally indistinguishable from the real blinding execution, with distinguishing advantage at most

$$(B.2) \quad N_{\text{PoK}}^{\text{zk}} \cdot \varepsilon_{\text{ABDLOP}}^{\text{zk}} + (W_{\max} + 2)t \varepsilon_{\text{hide}} + W_{\max} \varepsilon_{\text{hide},0} + \mathcal{O}(W_{\max} 2^{-\lambda}),$$

where $N_{\text{PoK}}^{\text{zk}} := 4 + 2t$ counts the proof-of-knowledge invocations that $\mathcal{S}$ must *simulate* rather than execute honestly, namely Steps 7, 12, 15 and 18 of Protocol 6 together with the t invocations of Step 1 and the t of Step 17 of Protocol 7. Step 1 of Protocol 6 is deliberately *not* among them as it concerns the blinding pairs, whose openings $\mathcal{S}$ generated itself at Step 2 of Protocol 19 and on which it therefore runs the honest prover, and likewise for Step 5 of Protocol 8. Here $\varepsilon_{\text{ABDLOP}}^{\text{zk}}$ is the Extended-MLWE advantage against the honest-verifier simulator of a single such invocation (hybrid H_1), and $\varepsilon_{\text{hide}}$ is the MLWE hiding advantage for a single ABDLOP commitment at the ambient (ℓ, m_2) (hybrids H_2 – H_4), $\varepsilon_{\text{hide},0}$ being the corresponding advantage for $C_{\mathbf{y},0}$, which is a tuple of $\tilde{N}$ blocks at those same parameters and so contributes $\tilde{N} \varepsilon_{\text{hide}}$ (Remark 4.21). (Recall that $\varepsilon_{\text{ABDLOP}}^{\text{zk}}$ is a *simulator* advantage not a knowledge-soundness errors $\varepsilon_{\text{ABDLOP}}^{(1)}, \varepsilon_{\text{ABDLOP}}^{(2)}$ of Proposition 4.17; in particular $N_{\text{PoK}}^{\text{zk}}$ is far smaller than $N_{\text{PoK}}^{(1)}$, which counts every invocation.) The attempt cap of Protocol 7 is what makes the coefficient $W_{\max} \leq 2^8$ a fixed quantity rather than an expectation.

- (2) $\mathcal{S}$ is EPT.
(3) $\mathcal{S}$'s output pair always lies in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$ (Definition 3.25).
(1) We proceed, similarly to the hybrid simulator construction in Lemma 3.6. Throughout, W denotes the blinding pairs of Step 2 together with their salts, and $\tilde{\rho}$ the accepting challenge block.

- H_0 : we define it to be the real experiment, that is, the real setting of LatticeBlindFold.
- H_1 : replace by the corresponding simulator output every sub-protocol transcript of Π'_{R1CS} (the linear, quadratic and anchoring ABDLOP proofs of knowledge, the commitment to coeffs, the Sum-Check round polynomials and h , and the transmitted σ), together with the t invocations of Step 1 of Protocol 7 on $\{C'_{1,j}\}_{j \in [t]}$ and the t invocations of its Step 17 on $\{C_{\mathbf{y},j}\}_{j \in [t]}$. The Sum-Check round polynomials, h and σ cost nothing here, being reproduced *exactly* rather than approximately (§4.1.1.1), so the only advantage incurred is that of the proofs of knowledge. Hybridising over the $N_{\text{PoK}}^{\text{zk}} = 4 + 2t$ invocations so replaced,

$$|H_1 - H_0| \leq N_{\text{PoK}}^{\text{zk}} \cdot \varepsilon_{\text{ABDLOP}}^{\text{zk}},$$

by the honest-verifier zero-knowledge of each sub-protocol (Lemma 3.6, Lemma 3.7, Lemma 3.16). The transmitted σ costs nothing here, since by Remark 4.2.(7) it is perfectly uniform in $\mathbb{K}$, so $\mathcal{S}$ reproduces it by sampling uniformly. Note that, after this process, the real witness $\mathbf{z}_1$ enters the view only through:

- the commitments $C'_{1,j} = \text{Commit}(y'_{1,j}, \mathbf{s}'_{1,j})$,
- the mask round, via $\mathbf{v} = \rho_1 \mathbf{z}_1 + \sum_i \rho_{K+i} \mathbf{z}_i^{\text{blind}}$ and $\mathbf{z} = \mathbf{v} + \mathbf{y}$, and
- the fixed public $c_1, \mathbf{x}_1$, which Definition 2.11 supplies to $\mathcal{S}$ and which are therefore not simulated here; we discharge them separately in at the end.
- H_2 : replace every *rejected*-attempt commitment $C_{\mathbf{y},0}^{(\tau)}, \{C_{\mathbf{y},j}^{(\tau)}\}_{j \in [t]}$, $\tau < \tilde{\tau}$, by commitments to 0 under fresh salts. None of them is ever opened, so

$$|H_2 - H_1| \leq (\tilde{\tau} - 1)(t \varepsilon_{\text{hide}} + \varepsilon_{\text{hide},0}) \leq W_{\max}(t \varepsilon_{\text{hide}} + \varepsilon_{\text{hide},0})$$

by the hiding of Commit (§2.5.2, MLWE); cf. Remark B.9. The two advantages are kept apart because $C_{\mathbf{y},0}^{(\tau)}$ is a tuple of $\tilde{N}$ commitments under independent salts rather than a single one, so that a further hybrid over its blocks gives $\varepsilon_{\text{hide},0} = \tilde{N} \varepsilon_{\text{hide}}$; see Remark 4.21. What is invoked here is plain MLWE hiding of $(\mathbf{t}_A, \mathbf{t}_B)$ and nothing more, since $C_{\mathbf{y},0}$ carries no proof of knowledge whose transcript would have to be simulated. The attempt cap is what turns the first bound, which is a random variable, into the second, which is a fixed quantity with $W_{\max}$.

- H_3 : replace $C'_{1,j}$ by $\text{Commit}(y'_{1,j}, \mathbf{s}'_{1,j})$ for the fake $\hat{\mathbf{z}}_1$ of Step 3, and derive $C_{\mathbf{y},j} := \text{Commit}(y_j, \mathbf{s}_j) - \sum_i \rho_i C'_{i,j}$. This is legitimated by the same identity already which holds in H_2 ($\mathcal{V}$'s Step 21 check,

satisfied by an honest prover), so $C_{\mathbf{y},j}$ was already a deterministic function of the other commitments, and after H_1 no opening of $C'_{1,j}$ is used anywhere in the view (in particular the Step 1 proof of knowledge of Protocol 7, the only place such an opening would be used, has already been replaced by $\mathcal{S}_{\text{ABDLOP}}$ there). The salts $s'_{1,j}$ are freshly sampled by the honest prover at Step 9 of Protocol 6 and by $\mathcal{S}$ at Step 4, so MLWE hiding applies verbatim and

$$|H_3 - H_2| \leq t \varepsilon_{\text{hide}}.$$

- H_4 : replace $\mathbf{z}_1$ by $\hat{\mathbf{z}}_1$ in the computation of $\mathbf{v}$, hence of $\mathbf{z}$. This hybrid simulator is Protocol 19. We claim

$$|H_4 - H_3| \leq t \varepsilon_{\text{hide}} + \mathcal{O}(W_{\max} 2^{-\lambda}).$$

Indeed:

- Both $\mathbf{v}$ -values lie in the set V of Lemma 3.20: $\|\mathbf{z}_1\|_\infty, \|\hat{\mathbf{z}}_1\|_\infty, \|\mathbf{z}_i^{\text{blind}}\|_\infty < b$, so $\|\mathbf{v}\|_\infty \leq (K+k)T(\mathcal{C})(b-1)$ in both experiments. Fix $(\tilde{\tau}, \tilde{\rho})$. This fixes $\mathbf{v}$ in each experiment. By Lemma 3.20, conditioned on acceptance, $\mathbf{z}$ is within $2^{-\lambda}$ of $D_{\mathfrak{s}}^{n_{\mathfrak{F}}}$ for every $\mathbf{v} \in V$, and the acceptance probability satisfies $p(\mathbf{v}) \in [(1-2^{-\lambda})/M, 1/M]$. Hence for any $\mathbf{v}, \mathbf{v}' \in V$ the distribution of the accepting index, namely a geometric distribution $\text{Geom}(p(\cdot))$ truncated at $W_{\max}$, satisfy $\Delta \leq \mathcal{O}(W_{\max} 2^{-\lambda})$, and likewise for the exhaustion event. Therefore $\tilde{\tau}$ (as well as the occurrence or not of $\mathcal{V}$'s $\perp$) is witness-independent up to $\mathcal{O}(W_{\max} 2^{-\lambda})$. Consequently $(\tilde{\tau}, \tilde{\rho}, \mathbf{z})$ is, in both experiments, within $\mathcal{O}(W_{\max} 2^{-\lambda})$ of the same law.
 - *Everything* the verifier sees afterwards is the same deterministic function of $(\tilde{\tau}, \tilde{\rho}, \mathbf{z})$ and of the fresh salt randomness (identically distributed in both). In particular $c_{\mathbf{y}} = \mathcal{L}(\mathbf{z}) - \rho_1 c_1 - \sum_i \rho_{K+i} c_i^{\text{blind}}$ and $\mathbf{x}_{\mathbf{y}} = \mathcal{L}_{\text{in}}(\mathbf{z}) - \rho_1 \mathbf{x}_1 - \sum_i \rho_{K+i} \mathbf{x}_i^{\text{blind}}$, where in H_3 both identities hold *automatically* (as $c_{\mathbf{y}} = \mathcal{L}(\mathbf{y})$, $\mathbf{z} = \mathbf{v} + \mathbf{y}$, $c_1 = \mathcal{L}(\mathbf{z}_1)$), and in H_4 the first holds by construction at Step 7(e) and the second because $\mathcal{L}_{\text{in}}(\hat{\mathbf{z}}_1) = \mathbf{x}_1$. Likewise $y_j = \widetilde{M_j \mathbf{z}(r)}$, $\text{Commit}(y_j, \mathbf{s}_j)$, the folded salts, and the whole Π'_{DEC} output $(\mathbf{z}'_i, y'_{i,j}, \mathbf{s}'_{i,j}, c'_{i,j}, \mathbf{x}'_i, C'_{i,j})$ are the same functions of $(\mathbf{z}, \text{salts})$ in both.
 - The single remaining difference is the message committed in the accepting round's $C_{\mathbf{y},j}$, namely $\widetilde{M_j(\mathbf{z} - \mathbf{v})(r)}$ versus $\widetilde{M_j(\mathbf{z} - \hat{\mathbf{v}})(r)}$. These commitments are never opened, the Step 17 proof of knowledge about them having been replaced by $\mathcal{S}_{\text{ABDLOP}}$ at H_1 , giving the stated $t \varepsilon_{\text{hide}}$.
- (2) Finally, membership in $\text{CE}_{\text{com}}(b, \mathcal{L}, \tilde{B}, \text{Commit}, T, K)^k$. $\|\mathbf{z}\|_\infty \leq \tau_{\lambda, \xi} \mathfrak{s} < B$ holds up to $2^{-\lambda}$ by Lemma 3.19 of §3.6, hence split_b yields $\|\mathbf{z}'_i\|_\infty < b$. The salt bound $\|\mathbf{s}'_{l,i,j}\|_\infty < \tilde{B}$ holds by construction, as well as $c'_i = \mathcal{L}(\mathbf{z}'_i)$, $y'_{i,j} = \widetilde{M_j \mathbf{z}'_i(r')}$, $C'_{i,j} = \text{Commit}(y'_{i,j}, \mathbf{s}'_{i,j})$ hold by construction.

It remains to remove the input instance from the simulator's input. Define $\mathcal{S}'(\text{pp}, \mathbf{s}, \mathbf{x}_1)$ to sample $c'_1 \xleftarrow{\mathfrak{s}} \mathbb{C}$, run $\mathcal{S}(\text{pp}, \mathbf{s}, (c'_1, \mathbf{x}_1))$ of Protocol 19, and output $((c'_1, \mathbf{x}_1), \mathbf{st}', \mathbf{w}')$. Two further hybrid simulators close the gap between H_4 and $\mathcal{S}'$.

- H_5 : hand $\mathcal{S}$ the real c_1 and compare against the real execution, now including u_1 in the compared tuple. This costs nothing beyond H_0 – H_4 . $\mathcal{S}$ is well defined on an arbitrary $c_1 \in \mathbb{C}$, consuming it only through the linear assignment $c_{\mathbf{y}} := \mathcal{L}(\mathbf{z}) - \rho_1 c_1 - \sum_{i \in [k]} \rho_{K+i} c_i^{\text{blind}}$ at Step 7(e), which requires no short preimage of c_1 , and u_1 is identical on both sides.
- H_6 : replace c_1 by the uniform c'_1 . After H_5 the transcript is a function of c_1 alone and not of $\mathbf{z}_1$, so it suffices that c_1 itself be close to uniform on $\mathbb{C}$.

This is the one step of the whole argument that depends on where the input relation is cut, and we isolate what it needs. Write $\mathbf{z}_1 = [\mathbf{z}_{\text{circ}}, \mathbf{e}]$ for the splitting induced by $\mathcal{I}_{\text{bl}}$. Over $\mathcal{R}' = \text{CCS}^\circ(b)$, $\mathcal{A}$ fixes $\mathbf{z}_{\text{circ}} = [x_1, \mathbf{w}_{\text{circ}}]$ (i.e. the whole of its output) while $\mathbf{e} \leftarrow \chi_b^{n_{\text{R}, \text{bl}}}$ is drawn afterwards by the prover inside ι_{bl} , from randomness independent of $\mathcal{A}$'s view. Lemma 3.3 then applies verbatim, with $A_{\text{circ}} \mathbf{z}_{\text{circ}}$ treated as an arbitrary adversarial shift, and gives

$$|H_6 - H_5| \leq \varepsilon_{\mathcal{L}\text{-hide}}.$$

Under regime (1) this step is statistical, and the bound holds for *every* $\mathbf{z}_{\text{circ}}$, so no distributional hypothesis on $\mathcal{A}$ is needed. Note also that ι_{bl} is total (Remark 3.2), so conditioning on the padded pair lying in $\text{CCS}(b)$ does not bias $\mathbf{e}$. Note finally that this hybrid is where c_1 is produced by $\mathcal{S}$

rather than received. Over the unpadded relation $u_2 = x_1$, so c_1 is a prover message and belongs to the compared st.

Summing H_0 – H_6 gives Equation (4.6) and completes the proof of Theorem 4.13.

We conclude this section with a series of remarks which should clarify some subtleties.

Remark B.8 (Soundness is not endangered by the late binding). *It is worth stating why the ability of $\mathcal{S}$ to fix c_y after $\bar{\rho}$ does not hand the same power to a cheating prover, since c_y is precisely the slack of the folded $\mathbb{C}$ -channel. In the protocol c_y is sent before $\bar{\rho}$; for a fixed c_y , a prover able to exhibit a short $\mathbf{z}$ with $\mathcal{L}(\mathbf{z}) = \rho_1 c_1 + \sum_i \rho_{K+i} c_i^{\text{blind}} + c_y$ for two distinct challenge blocks agreeing outside one coordinate yields, by the extraction of Lemma 4.10, a short opening of the corresponding input commitment, i.e. breaks relaxed binding of $\mathcal{L}$ (MSIS). The simulator escapes this only because Definition 2.11 lets it choose ω' , hence $\bar{\rho}$, itself: it is never confronted with a random challenge after committing. This is the usual asymmetry behind HVZK, and it is also the reason both c_y and $\mathbf{x}_y$ must be bound by a pre-challenge message in Protocol 7. Only their disclosure may be moved after $\bar{\rho}$ (Remarks B.3 and B.9).*

Remark B.9 (Rejected rounds are public). *In the interactive setting a rejected attempt of the Rej_1 loop of Protocol 7 is not discarded and it remains part of $\mathcal{V}^*$'s view. This matters, because Lemma 3.20 controls the distribution of $\mathbf{z}$ only conditioned on acceptance. Conditioned on rejection, $\mathbf{y}$ is distributed as $D_{\mathbb{S}}^{\text{re}}$ reweighted by the complement of the acceptance probability, which depends on $\mathbf{v} = \sum_i \rho_i \mathbf{z}_i$ and hence on the genuine witness $\mathbf{z}_1$. Likewise, the simulator $\mathcal{S}_{\text{ABDLOP}}$ of Lemma 3.6 produces non-aborting transcripts only.*

This is why the pair $(c_y, \mathbf{x}_y) = (\mathcal{L}(\mathbf{y}), \mathcal{L}_{\text{in}}(\mathbf{y}))$ is committed rather than sent inside the loop. Since $\mathcal{L}$ is a plain Ajtai commitment, it is not hiding (Remark 2.29), and carries no salt able to absorb the conditioning above. An in-loop c_y would therefore expose a $\mathbf{v}$ -dependent quantity on every failed attempt. With the commitment, each attempt transmits only $C_{y,0}$ and $\{C_{y,j}\}_{j \in [t]}$, all of them ABDLOP commitments under freshly sampled salts and hence hiding under MLWE (§2.5.2). Note that in the argument of Theorem 4.13 they are replaced by commitments to 0, at cost $t \cdot \varepsilon_{\text{hide}} + \varepsilon_{\text{hide},0}$ per attempt, the two advantages being those of two different ABDLOP instances (Remark 4.21). Over the W attempts of the loop this contributes $W \cdot (t \varepsilon_{\text{hide}} + \varepsilon_{\text{hide},0})$, with $W = \mathcal{O}(1)$ in expectation.

Note that $c_y \in \mathbb{R}_{\mathbb{F}}^{\kappa}$ and $\mathbf{x}_y \in \mathbb{R}_{\mathbb{F}}^{n_{\text{R},\text{in}}}$ are $\mathbb{R}_{\mathbb{F}}$ -vectors and are committed in BDLOP message slots. Being already defined over $\mathbb{R}_{\mathbb{F}}$, this requires none of the rank-doubling of §3.3. Its salt block is kept disjoint from those of $\{C_{y,j}\}$ precisely because $C_{y,0}$ is the one commitment opened in the clear, and an ABDLOP opening discloses $\mathbf{s}_2$. Their joint length $\kappa + n_{\text{R},\text{in}}$ exceeds the ℓ message slots of a single commitment, so $C_{y,0}$ is a tuple of $\tilde{N}$ blocks at the ambient parameters, each with a salt pair of its own (Remark 4.21); the hiding advantage it contributes to Equation (4.6) is correspondingly $\varepsilon_{\text{hide},0} = \tilde{N} \varepsilon_{\text{hide}}$.

Remark B.8 and the extractor of Lemma 4.10 require that c_y is fixed before $\bar{\rho}$ is drawn, not that it be disclosed before $\bar{\rho}$. $\mathcal{V}$ does not consume c_y until it folds $c \leftarrow \sum_i \rho_i c_i + c_y$, after the loop has terminated. Binding of Commit pins c_y at the moment $C_{y,0}$ is sent, so a cheating prover cannot adapt it to $\bar{\rho}$. The norm check on the revealed salts is what supplies this. Without it, the commitment is not binding in any sense, since for any $\Delta \mathbf{s}_2 \neq 0$ one may solve $A_1 \Delta \mathbf{s}_1 = -A_2 \Delta \mathbf{s}_2$ and open to $c_y - B \Delta \mathbf{s}_2$, which is exactly the late binding that Remark B.8 reserves for the simulator. With the check, two accepting openings of $C_{y,0}$ to distinct pairs $(c_y, \mathbf{x}_y) \neq (c'_y, \mathbf{x}'_y)$ differ in some block, and that block yields $A_1 \Delta \mathbf{s}_1 + A_2 \Delta \mathbf{s}_2 = 0$ with $0 \neq \|(\Delta \mathbf{s}_1, \Delta \mathbf{s}_2)\|_{\infty} < 2\tilde{B}$, i.e. a solution to MSIS $_{\kappa, m_1+m_2, 2\tilde{B}\sqrt{(m_1+m_2)d}}$ at the ambient salt width; hence across rewind branches the extractor of Lemma 4.10 recovers the same c_y and the same $\mathbf{x}_y$, except with probability $\epsilon_{\text{bind}}(\tilde{B})$, which is what Remark B.3 needs. In fact this one commitment is statistically binding on the revealed-salt set (for suitable parameters), so the MSIS appeal is conservative. The count is over differences rather than over salt pairs, and it is read blockwise, at the ambient width: at $d = 128$, $\kappa = 11$ and $m_1 + m_2 = 68$ there are at most $5^{(m_1+m_2)d}$ candidate differences against $q^{\kappa d/l} = 2^{45056}$ target values, $l = 2$ being the number of irreducible factors of $X^d + 1$ modulo q , so a uniform $[\mathbf{A}_1 \mid \mathbf{A}_2]$ kills none of them except with overwhelming probability; the $d = 64$ set reads the same way, at $q^{\kappa d/l} = 2^{38912}$ against $m_1 + m_2 = 98$. Equation (4.19) is the general form, and it holds for every circuit, the width being independent of $n_{\text{R},\text{in}}$ (Remark 4.21). We nonetheless state the argument computationally, since that is the form the composition consumes. Since the salts are opened in the clear, the verifier reads the bound off directly, at cost $\mathcal{O}(\tilde{N}(m_1 + m_2)d)$ and with no appeal to the norm-proving machinery of §3.3.4, which is needed only for salts that remain hidden. Recall that this is also the case for $\{C_{y,j}\}_{j \in [t]}$, whose bounds

continue to be enforced in relaxed form by the proof of knowledge at Step 17 and by $\mathcal{R}_2$ -membership. As with the input salts of Protocol 8, this suffices for a single invocation of Π_{LBF} but not for unbounded iteration of $\Pi_{\text{LBF-Fold}}$, where the mask salts enter the folded salt of every round; cf. Remark 4.8. Only the opening of $C_{\mathbf{y},0}$, which carries both $c_{\mathbf{y}}$ and $\mathbf{x}_{\mathbf{y}}$, is processed after $\vec{\rho}$; both values are already bound before it. The simulator also samples ω' itself (Remark B.7.(2)) and so knows $\vec{\rho}^{(\tau)}$ before producing any prover message.

With the attempt cap of Protocol 7 in force, the number of rejected attempts appearing in $\mathcal{V}^*$'s view is bounded by $W_{\max}$ rather than merely finite, so the accounting below is over at most $W_{\max}$ terms and the bound of Theorem 4.13 may be read with the fixed coefficient $W_{\max} \leq 2^8$ in place of a random attempt count.

Remark B.10 (The simulator is interactive-only). The simulator of Protocol 19 rests on $\vec{\rho}^{(\tau)}$ being a deterministic function of ω' rather than of the transcript, which is exactly how Definition 2.11 models an honest-but-curious public-coin $\mathcal{V}^*$. However, under the Fiat–Shamir instantiation we first discussed in Remark 4.2.(2) this is false. There, $\vec{\rho}^{(\tau)} = \text{hash}(\text{transcript} \| C_{\mathbf{y},0}^{(\tau)})$, so computing $c_{\mathbf{y}}$ would require $\vec{\rho}^{(\tilde{\tau})}$, which requires $C_{\mathbf{y},0}^{(\tilde{\tau})}$. In order to reconcile the two settings, in the random-oracle model, the simulator instead needs to programs hash , once per attempt. More precisely, it draws $\vec{\rho}^{(\tau)}$, computes the attempt as at Step 7, and then sets $\text{hash}(\text{transcript} \| C_{\mathbf{y},0}^{(\tau)}) := \vec{\rho}^{(\tau)}$, aborting on the negligibly likely event that the point has already been queried. Since at most $W_{\max}$ attempts occur, at most $W_{\max}$ programming steps are needed and the abort probability remains negligible.

The resulting statement is a ROM one and should not be conflated with the standard-model statement of Theorem 4.13. Note finally that Fiat–Shamir simplifies the treatment of aborts, since rejected attempts are never output and the $W_{\max}(t+1)\varepsilon_{\text{hide}}$ term of H_2 disappears, while complicating the simulator.