SALSAA – Sumcheck-Aided Lattice-based Succinct Arguments and Applications

Shuto Kuriyama, Russell W. F. Lai, Michał Osadnik, and Lorenzo Tucci

Aalto University, Espoo, Finland
{shuto.kuriyama, russell.lai, michal.osadnik, lorenzo.tucci}@aalto.fi

Abstract. We present SALSAA, a more efficient and more versatile extension of the state-of-the-art lattice-based fully-succinct argument frameworks, "RoK, paper, SISsors (RPS)" and "RoK and Roll (RnR)" [Klooß, Lai, Nguyen, and Osadnik; ASIACRYPT'24, '25], integrating the *sumcheck* technique as a main component. This integration enables us to design an efficient norm-check protocol (controlling the norm during witness extraction) with a strictly linear-time prover while reducing proof sizes by 2–3× compared to the previous quasi-linear-time norm-check in RPS/RnR, eliminating a central performance bottleneck. The sumcheck integration also allows us to natively support a wider class of relations, including rank-1 constraint systems (R1CS), which are widely used to express real-world computations.

To demonstrate the versatility and efficiency of our framework, we showcase three impactful applications achieved by different RoKs (Reductions of Knowledge) compositions: (i) a lattice-based *succinct argument of knowledge* with a linear-time prover, achieving a verifier time of 41 ms, prover runtime of 10.61 s, and proof size of 979 KB for a witness of 2^{28} $\mathbb{Z}_q$ elements; (ii) a *polynomial commitment scheme* with matching performance; and (iii) the first *lattice-based folding scheme* natively operating on ℓ_2-norm-bounded witnesses, achieving highly efficient verification in 2.28 ms and producing a proof of just 73 KB for a witness of 2^{28} $\mathbb{Z}_q$ elements, outperforming prior works for the family of linear relations.

We provide a *modular, concretely efficient* Rust implementation of our framework, benchmarked over cyclotomic rings with AVX-512-accelerated NTT-based arithmetic, demonstrating the practical efficiency of our approach.

1 Introduction

Succinct argument systems [Kil92, Mic94] are cryptographic protocols that enable a prover to convince a verifier of the validity of a mathematical statement with communication much shorter than the size of the underlying data. Lattice-based argument systems, in particular, offer the promise of post-quantum security, unlike group-based constructions (e.g. [PHGR13, Gro16, BBB+18, CHM+20]), and can use rich algebraic structures that may lead to greater efficiency and expressiveness compared to hash-based approaches (e.g. [BBHR18, BCR+19, GLS+23, AHIV23]). This work contributes to the ongoing effort to realise this potential in practice.

Succinct non-interactive arguments of knowledge (SNARKs) are a central component in scenarios requiring efficient verification such as verifiable delegation of computation [KP16, Lee21, GSW23], verifiable fully homomorphic encryption [GBK+24, ABPS24, LLZ+25, ZLW+25], aggregate signatures [BK20, JRS23, JRS24, AAB+24, DKLW25], and verifiable delay functions [LM23, OKC+25].

Split-and-fold arguments. A central paradigm in lattice-based succinct arguments is the *split-and-fold* approach, introduced in [BLNS20] and inspired by Bulletproofs [BBB+18]. In this framework, the prover recursively reduces a high-dimensional inhomogeneous short integer solution (ISIS) relation to lower-dimensional ones, eventually reaching a size where the witness can be revealed directly. This strategy underlies many recent works [AL21, BCS21, ACK21, CLM23, BS23, NS24, AFLN24, CMNW24, KLNO24, KLNO25], including fully-succinct arguments [CLM23, AFLN24, CMNW24, KLNO24, KLNO25] that achieve verification times significantly faster than checking the relation naively, potentially after an offline preprocessing phase.

Folding schemes. Orthogonal to the aforementioned "monolithic" (following the terminology e.g. from [BCFW25]) SNARKs, a line of works [BC25a, FKNP24, BMNW25a, BMNW25b, NS25, BC25b, BCFW25]

have proposed quantum-secure folding schemes (or accumulation schemes) which can be used as building blocks of incrementally-verifiable computation protocols (IVC) [Val08] and proof-carrying data (PCD) [CT10]. The prover in an IVC generates a proof for small steps of the entire computation and proves the smaller computation step by step, while the prover in monolithic SNARKs proves the entire computation in one shot. An IVC was originally constructed from the recursive composition of monolithic SNARKs [Val08, BCTV14], however, this construction requires the verifier circuit for the previous proofs to be included in the statement at every computation step which introduces considerable overhead. An idea of folding scheme was introduced [BGH19] as an alternative approach to construct a highly-efficient IVC. The prover of an IVC accumulates the proof of each step in an accumulator and then the verifier runs a single expensive check at the end of the protocol, which is more efficient than running the verifier at each step.

Among recent quantum-secure folding schemes, some of them are based on lattices [BC25a, FKNP24, NS25, BC25b], yet each of those schemes concerns mainly the coefficient ℓ_∞-norm and supports only low-norm witnesses (or rather, scales poorly with the norm growth). None of those SNARKs yields native support for the ℓ_2-norm. Further, even state-of-the-art lattice-based folding scheme [BC25b] are not demonstrated to be concretely efficient. As an example, the prover in [BC25b] has to compute a "double commitment" of the decomposed witness vector at each step of the folding, which is an order of magnitude more expensive than a "single" Ajtai commitment.

"RoK, Paper, SISsors (RPS)" and "RoK and Roll (RnR)". The recent works of "RoK, paper, SISsors" (RPS) and "RoK and Roll (RnR)" [KLNO24, KLNO25], building upon the *vanishing short integer solution* (vSIS) assumption [CLM23, JL25], provide a suite of general-purpose tools for fully-succinct lattice-based arguments. More precisely, the tools in [KLNO24, KLNO25] are described in the language of reduction of knowledge (RoK) [KP23], where an RoK is an argument system allowing a prover and a verifier to jointly reduce an instance $(\mathsf{stmt}, \mathsf{wit})$ of some relation Ξ to an instance $(\mathsf{stmt}', \mathsf{wit}')$ of another (supposedly simpler) relation Ξ'. Roughly, the RoKs in [KLNO24, KLNO25] reduce to structured Inhomogeneous Short Integer Solution (ISIS) relations, denoted Ξ^{lin}, whose statements are structured linear equations and witnesses are low-norm vectors satisfying these equations.

RoK and Roll [KLNO25] improves upon [KLNO24] by introducing a new RoK which computes and proves about randomised projections of the witness. They can be seen as structured/improved variants of the randomised projection steps used in LaBRADOR [BS23]. This allows RnR to use a much larger challenge set in the folding step Π^{fold} instead of a subtractive set (used in [KLNO24]), which ultimately removes (roughly) one λ factor from the proof size of [KLNO24].

A central tool in both RPS and RnR is a fully-succinct norm-check RoK[1] $\Pi^{\mathsf{norm\text{-}klno24}}$ which reduces Ξ^{lin} with an exact norm claim (i.e. without soundness gap) to Ξ^{lin} with a relaxed norm claim (i.e. with soundness gap).[2] Composing a number of atomic RoK protocols including $\Pi^{\mathsf{norm\text{-}klno24}}$ results in the fully-succinct argument frameworks RPS and RnR for Ξ^{lin} with no soundness gap.

However, the norm-check RoK $\Pi^{\mathsf{norm\text{-}klno24}}$ becomes the bottleneck when it comes to the efficiency of the RPS and RnR frameworks. The prover in the norm-check RoK $\Pi^{\mathsf{norm\text{-}klno24}}$ is asymptotically quasi-linear in the witness size. This is because the prover needs to compute the multiplications of two polynomials with the degree corresponding to the size of the witness. This translates to a concretely high prover runtime for the resulting argument system. Norm-check RoK $\Pi^{\mathsf{norm\text{-}klno24}}$ also incurs the overhead on the proof size since the prover needs to further commit to the result of the aforementioned polynomial multiplication and consider it as a part of the witness throughout the protocol.

Papercraft. Apart from the theoretical interest, the frameworks of RPS and RnR have been designed with practical efficiency in mind, especially with respect to the verifier runtime and proof size. Papercraft [OKC+25] can be seen as a preliminary implementation of RPS which constructed a verifiable delay function (VDF) by combining the framework of RPS with the lattice-based sequential function candidate of [LM23]. This prototype implementation demonstrated the practical feasibility of the RPS framework.

However, the implementation of Papercraft [OKC+25] is limited in the following aspects: (i) It only showcases a specific application (VDF) and does not provide a general-purpose toolset which can be used for building applications, such as more general-purpose SNARKs, polynomial commitment schemes (PCS)

[1] Called Π^{norm} in [KLNO24], renamed to $\Pi^{\mathsf{norm\text{-}klno24}}$ here.

[2] For the canonical ℓ_2-norm defined over a cyclotomic ring.

or folding schemes. (ii) It relies on a very small challenge set inheriting from the RPS [KLNO24], which limits the choice of parameters and results in larger proof sizes (the issue was resolved in [KLNO25]). (iii) It does not incorporate optimisations such as hardware acceleration or even NTT-based ring operations.

1.1 Our Contributions

In this work, we further improve and extend the RPS [KLNO24] and the RnR [KLNO25] frameworks to a new framework, `SALSAA`, for constructing efficient and versatile lattice-based succinct arguments. On the technical level, we introduced the sumcheck technique to our frameworks (RPS, RnR), which primarily enables us to construct an efficient norm-check protocol, one of the most computationally intensive RoKs in RPS and RnR. Our norm-check has 2-3× smaller proof size compared to the norm-check in RPS and RnR, and its prover runs in strict linear-time rather than quasi-linear time. This mitigates the key efficiency bottlenecks of RPS and RnR, as experienced in Papercraft [OKC+25].

To illustrate the versatility and efficiency of our framework, we demonstrate concretely efficient applications of `SALSAA` including a SNARK, a polynomial commitment scheme (PCS), and a folding scheme. We provide an efficient and general-purpose implementation of `SALSAA`, which can be used in high-level applications such as those listed above. Finally, we benchmark the performance of our framework in each application (a SNARK/PCS and a folding scheme) and compare them with competing solutions in the literature. We outline our contributions in more detail below.

Norm-check. We design an alternative norm-check RoK Π^{norm} based on the sumcheck protocol [LFKN92] reducing a norm relation Ξ^{norm} to a sumcheck relation $\Xi^{\overline{\mathsf{sum}}}$. The main idea is to express a claim about the norm of the witness vector $\mathbf{w}$ as a sumcheck claim of a low-degree extension of $\mathbf{w}$. Inheriting the linear-time[3] prover of the sumcheck, the prover of the norm-check runs in time linear in the witness size, whereas the norm-check RoK $\Pi^{\mathsf{norm\text{-}klno24}}$ of [KLNO24, KLNO25] requires quasi-linear time. This is because the prover operation in $\Pi^{\mathsf{norm\text{-}klno24}}$ involves the multiplication of two polynomials of degree m, resulting in $\mathcal{O}(m \log m)$ complexity. The output of this multiplication is further committed as additional columns of the witness. Replacing this step with a simple evaluation claim as a result of the sumcheck reduction (without alternating the witness) yields an improvement of 2-3× factor[4] in the proof size compared to RnR.

As a part of the reduction from Ξ^{norm} to Ξ^{lin}, we provide a low-degree extension (LDE) relation Ξ^{lde} and its structured variant $\Xi^{\mathsf{lde}\text{-}\otimes}$ which extend the Ξ^{lin} relation of [KLNO24, KLNO25] to additionally check that the LDE of the witness evaluates to specified values at specified points. We also formalise a sumcheck relation $\Xi^{\overline{\mathsf{sum}}}$, which similarly extends Ξ^{lin} to check a sumcheck claim over LDEs and which we reduce to the $\Xi^{\mathsf{lde}\text{-}\otimes}$ relation. All relations other than Ξ^{lin} considered in this work, e.g. including Ξ^{lde}, $\Xi^{\overline{\mathsf{sum}}}$ and $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ (variants of $\Xi^{\overline{\mathsf{sum}}}$ formalised in Appendix B), can be reduced to $\Xi^{\mathsf{lde}\text{-}\otimes}$, which is further reduced to Ξ^{lin}.

Applications (SNARK, PCS, Folding Scheme). We first show the asymptotically efficient composition of the RoKs from this work and [KLNO24, KLNO25] to construct an efficient SNARK. Subsequently, we provide the construction of a (multilinear/univariate) PCS built upon the obtained SNARK and a vSIS commitment scheme.

We construct a lattice-based folding scheme with a linear-time prover built on `SALSAA` (see Table 2) which achieves a proof size of 70 KB, without relying on the double commitment technique used in LatticeFold+ [BC25b]. It is instantiated with the composition of the RoKs Π^{join}, $\Pi^{\otimes\mathsf{RP}}$, Π^{fold}, $\Pi^{b\text{-}\mathsf{decomp}}$, Π^{norm}, $\Pi^{\overline{\mathsf{sum}}}$, $\Pi^{\mathsf{lde}\text{-}\otimes}$, Π^{batch^*} from this work as well as [KLNO24, KLNO25]. The proposed folding scheme can reduce multiple instances $\left(\Xi^{\mathsf{lin}}_{(i)}\right)^L$ into a single instance in Ξ^{lin}.

Besides the norm-check, the introduction of sumcheck technique allows us to build protocols that enhance the aforementioned applications, such as an efficient batching protocol Π^{batch^*} (Section 6.3) for a folding scheme, which is alternative to Π^{batch} in RPS and RnR, and a RoK from rank-1 constraint systems (`R1CS`) to Ξ^{lin} (Appendix C). The latter enables our framework to natively capture a wider range of relations in the context of both a folding scheme and a succinct argument of knowledge.

[3] Assuming that the sumcheck prover is optimised via the dynamic programming technique [Tha13, XZZ+19, CBBZ23].

[4] This can be confirmed by comparing the proof size of this work and that of [KLNO25] on Table 1 since the only difference between them is the norm-check.

Scheme	$\#\mathbb{Z}_q = 2^{26}$				$\#\mathbb{Z}_q = 2^{28}$				$\#\mathbb{Z}_q = 2^{30}$			
	Comm.	$\mathcal{P}$	$\mathcal{V}$	Pr. Size	Comm.	$\mathcal{P}$	$\mathcal{V}$	Pr. Size	Comm.	$\mathcal{P}$	$\mathcal{V}$	Pr. Size
Brakedown [GLS+23]	36s	3,21s	0.703s	49157 KB	150s	13s	2.56s	93767 KB	605s	48.6s	2.96s	181948 KB
Ligero [AHIV17]	39.9s	3,11s	0.196s	7256 KB	169s	12.4s	0.402s	14383 KB	717s	50s	0.846s	28631 KB
FRI [BBHR18]	168s	185s	0.041s	740 KB	-	-	-	-	-	-	-	
CMNW24 [CMNW24]	-	-	-	1546 KB	-	-	-	-	-	-	-	5296 KB
HSS24 [HSS24]	188s		1.07s	48640 KB	-	-	-	-	-	-	-	
Papercraft [OKC+25]	-	-	-	-	1593s		5.10s	18940 KB	6986s		6.00s	29790 KB
KLNO25 [KLNO25]	-	-	-	2181 KB	-	-	-	2604 KB	-	-	-	3152 KB
Greyhound [NS24]	4.37s	2.03s	0.492s	46 KB	21.2s	8.21s	1.15s	53 KB	132s	41.2s	2.80s	53 KB
This work ($\varphi = 128$)	0.080s	3.05s	0.034s	808 KB	0.348s	10.61s	0.041s	979 KB	1.23s	39.7s	0.054s	1123 KB
This work ($\varphi = 256$)	0.054s	2.89s	0.032s	1005 KB	0.211s	10.49s	0.045s	1232 KB	0.72s	37.95s	0.054s	1459 KB
This work ($\varphi = 512$)	0.034s	2.87s	0.033s	1370 KB	0.135s	10.87s	0.047s	1694 KB	0.48s	41.74s	0.063s	2018 KB

Table 1: Concrete performance evaluation of this work compared to prior post-quantum SNARKs/PCSs. Comm. is time spent during the commitment phase. Both running time and proof size of the prior works are taken from Tabel 1,2 in Greyhound [NS24]. Brakedown [GLS+23], Ligero [AHIV17], FRI [BBHR18] are hash-based schemes, and the rest are lattice-based schemes. The experiment of our work was performed on three different degrees of cyclotomic polynomials, which we denote by φ.

Concretely efficient implementation. We provide implementations of the RoKs introduced in this work, including a novel Π^{norm} protocol as well as RoKs from the prior works. Our implementation incorporates multiple state-of-the-art optimisations, including the use of AVX-512-IFMA instructions for hardware acceleration of cyclotomic ring arithmetic and parallelisation. Our parameters have been carefully selected, ensuring practical performance without compromising security, e.g. so that the cyclotomic ring splits only up to small-degree extension (also known as an "incomplete NTT" transformation [CHK+21, LSS+21, HYJ+25, PMH+25]). On the usability front, our codebase is designed with modularity in mind, allowing easy integration of new RoKs and applications. Nevertheless, we acknowledge that further work is needed to enhance the usability of our codebase and leave as future work e.g. providing a high-level domain-specific language (DSL) for defining relations, and compiling them into our framework, in manner similar to [LSS24]. Our code is made publicly available[5].

Our implementation includes the applications of our framework, specifically a succinct argument of knowledge, a PCS, and a folding scheme. We performed comprehensive benchmarks. Experimental results on the SNARK/PCS applications show a significant advantage over the existing state-of-the-art lattice-based schemes as well as hash-based schemes (Table 1) concerning especially the verifier runtime, while keeping the proof size and the prover runtime competitive[6]. This confirms that the asymptotic gains achieved by our protocol translate into practical improvements.

On the concrete side, our SNARK (and implied PCS) shall be viewed as *the first lattice-based SNARK with the verification below* 50 *ms and proof size below* 1 *MB for a witness size of* 2^{28} $\mathbb{Z}_q$ *elements*. This setting is particularly relevant for practical applications such as verifiable computation, where (usually) the verifying party is a client with limited computational resources, and the witness size is large (representing the trace of the computation). While the prover greatly benefits from extensive parallelisation across many cores, the verifier's code is mostly sequential and remains efficient even on systems with limited multithreading capabilities. Our folding scheme achieves a proof size of 73 KB with a verifier measured in less than 3 ms for folding 4 instances of a linear relation with a witness size of 2^{30} $\mathbb{Z}_q$.

[5]Open-source: `github.com/lattice-arguments/salsaa`.

[6]This design decision of prioritising the verifier runtime shall be viewed as complementary to the design of Greyhound [NS24], which prioritises the proof size and the prover runtime and we remark about the possibility of combining the techniques of both via hybrid approach.

1.2 Related Work

Group- and hash-based constructions used to dominate the literature on succinct arguments. Group-based argument systems such as Groth16 [Gro16], PLONK [GWC19], and others [PHGR13, GGPR13, WTs+18, MBKM19, XZZ+19, BFS20, CHM+20] achieve strong efficiency but rely on group-based assumptions like discrete logarithm, and are thus insecure against quantum adversaries. Hash-based argument systems, such as FRI [BBHR18], Ligero [AHIV23], Aurora [BCR+19], and Breakdown [GLS+23], offer post-quantum security and transparent setup. However, their lack of algebraic structure makes efficiency improvements increasingly difficult.

Lattice-based arguments, on the other hand, still have ample potential for efficiency improvements by tapping into currently underutilised algebraic structures. Lattice-based argument systems have evolved into a diverse landscape. Non-succinct lattice-based arguments [LNSW13, BLS19, ALS20, LNP22] have their proof size and verification time linear in the witness size, which offers high concrete efficiency for small relations. In a semi-succinct setting, the proof size is polylogarithmic in the witness size, while verification is not. LaBRADOR [BS23] achieves state-of-the-art proof sizes in this category, outperforming hash-based systems such as FRI [BBHR18], Breakdown [GLS+23], Ligero [AHIV23], and Aurora [BCR+19]. Greyhound [NS24], a lattice-based polynomial commitment scheme (PCS), achieves polylogarithmic proof size and sublinear (but not polylogarithmic) verification time, and compares favourably to both hash-based [BBHR18, AHIV23, GLS+23] and lattice-based PCS [CMNW24, HSS24]. A fully-succinct construction in the standard model was first proposed in [ACL+22] and later improved in [CLM23, FLV23], based on a lattice-based knowledge assumption which was later morally disproved in [WW23].[7] The independent works of [BCS23, CLM23] concurrently propose lattice-based Bulletproofs variants with polylogarithmic verification time. [CLM23] introduces a commitment scheme with polynomial structure based on a new assumption (vSIS), combined with structured folding to achieve succinct verification. In contrast, [BCS23] uses a delegation protocol inspired by [Lee21] to avoid linear verification time. Subsequent works [AFLN24, CMNW24] improves the theoretical parts by relying purely on standard lattice assumptions, while maintaining similar efficiency. The latest framework RPS and RnR [KLNO24, KLNO25] achieved the best asymptotic and concrete efficiency so far.

Polynomial interactive oracle proofs (PIOP), e.g. [BCG+19, GWC19, BFS20, CHM+20, BCHO22, XHY24], are powerful information-theoretic building blocks for constructing succinct arguments via compilation [BCS16] with polynomial commitment schemes (PCS), e.g. [CBBZ23, Lib24]. This approach of constructing succinct arguments allows a more modular construction based on different assumptions, combining different choices of PIOPs and PCSs. In this work, we draw inspiration from the multivariate sumcheck PIOP used in multiple previous works [CHM+20, Set20, CBBZ23, XZZ+19] (and adapted to univariate setting [BCHO22]) that allows efficient sumcheck claim verification. Deviating from the modular framework of compiling PIOPs with polynomial commitments, this work employs the sumcheck PIOP in a non-black-box way in conjunction with the lattice-based argument toolkit of [KLNO24, KLNO25].

1.3 Technical Overview

In this subsection, we provide a high-level overview of the technical contributions, mainly focusing on the novel norm-check RoK. The composition of RoKs for the purpose of constructing more advanced primitives, e.g. SNARKs and folding schemes, is relatively straightforward following the blueprint established in [KLNO24, KLNO25] and thus omitted here.

This work builds upon and extends the toolkit provided in [KLNO24, KLNO25] for building fully-succinct lattice-based arguments. Before explaining our results, we first provide a brief overview of the [KLNO24, KLNO25] toolkits.

Throughout, we denote by $\mathcal{K} = \mathbb{Q}(\zeta)$ a cyclotomic field with conductor $\mathfrak{f}$ of degree $\varphi = \varphi(\mathfrak{f})$, and its ring of integers by $\mathcal{R} := \mathcal{O}_{\mathcal{K}} = \mathbb{Z}[\zeta]$. Given the canonical embedding $\sigma : \mathcal{K} \to \mathbb{C}^{\varphi}$ over $\mathcal{K}$ and an element $x \in \mathcal{K}$, we write $\sigma(x) := (\sigma_j(x))_{j\in[\varphi]}$, where $\sigma_j \in \mathsf{Gal}(\mathcal{K}/\mathbb{Q})$. We extend the notation of σ naturally to vectors, i.e. if $\mathbf{x} = (x_i)_{i\in[m]} \in \mathcal{K}^m$, then $\sigma(\mathbf{x})$ is defined as the concatenation $(\sigma(x_i))_{i\in[m]}$.

Framework of RoK, paper, SISsors. [KLNO24, KLNO25] comprise a suite of reductions of knowledge (RoK) for algebraic relations defined over $\mathcal{R}$. A RoK is an interactive protocol that allows a prover with

[7] However, no attacks against the argument systems themselves are known.

input $(\mathsf{stmt}, \mathsf{wit})$, a statement and a witness respectively, and a verifier with input of the same statement stmt, to jointly agree on a new statement stmt' for which the prover knows the witness wit'. The knowledge soundness of the RoK guarantees that if the prover indeed knows the derived witness wit' of stmt', then it must also know a witness wit to the original statement stmt.

The toolkits enable the construction of fully-succinct arguments, a PCS, and a folding scheme for the principal relation Ξ^{lin} by iteratively composing different RoKs, reducing Ξ^{lin} instances to other Ξ^{lin} instances, e.g. shrinking dimensions. In brief, a Ξ^{lin} instance consists of a statement $(\mathbf{H}, \mathbf{F}, \mathbf{Y})$ and a witness $\mathbf{W} \in \mathcal{R}^{m \times r}$ which satisfy the relation $\mathbf{HFW} = \mathbf{Y} \bmod q$ and $\|\mathbf{W}\| \leq \beta$, where the matrix $\mathbf{F}$ has a row-tensor structure, i.e. it can be written as $\mathbf{F} = \mathbf{F}_0 \bullet \ldots \bullet \mathbf{F}_{\mu-1}$, where $\mathbf{F}_i \in \mathcal{R}_q^{n \times d}$ for $i \in [\mu]$ and $\bullet$ denotes the row-wise Kronecker product.[8] This property is denoted by the shorthand $\mathbf{F} \in \mathcal{R}_q^{n \times d^{\otimes \mu}}$. In this overview, we conveniently ignore that $\mathbf{W}$ is a matrix and treat it as a vector in $\mathcal{R}^m$, where $m = d^\mu$, as the generalisation is straightforward.

The vanishing short integer solution vSIS assumption (family) [CLM23] states that, for a random row-tensor $\mathbf{F} \leftarrow\!\!\$\ \mathcal{R}_q^{n \times d^{\otimes \mu}}$, it is hard to find a short vector $\mathbf{x}$ satisfying $\mathbf{Fx} = \mathbf{0} \bmod q$. Based on this assumption, for a short vector $\mathbf{w}$, the vector $\mathbf{y} = \mathbf{Fw} \bmod q$ is a computationally binding commitment (called the vSIS commitment) of $\mathbf{w}$. Depending on the tensor structure of $\mathbf{F}$, the product $\mathbf{Fw}$ can be seen as the evaluation of a polynomial with coefficients given by $\mathbf{w}$. For example, if $m = 2^\mu$ and $\underline{\mathbf{F}} = (1, f, \ldots, f^{m-1}) = (1, f^{m/2}) \otimes (1, f^{m/4}) \otimes \ldots \otimes (1, f)$, then $\underline{\mathbf{F}}\mathbf{w} = \sum_{i=0}^{m-1} w_i f^i$. By allocating the top rows $\overline{\mathbf{F}}$ of $\mathbf{F}$ for the vSIS commitment and the bottom rows $\underline{\mathbf{F}}$ for polynomial evaluations, the Ξ^{lin} relation can be seen as a polynomial commitment relation.

Low-Degree Extension Relation. We are particularly interested in a type of multivariate polynomial which is a low-degree extension of $\mathbf{w}$. In more detail, a degree-d extension of $\mathbf{w}$ is a polynomial $\mathsf{LDE}[\mathbf{w}] : \mathcal{R}^\mu \to \mathcal{R}$ satisfying

$$(\mathsf{LDE}[\mathbf{w}](\mathbf{z}))_{\mathbf{z} \in [d]^\mu} = (w_i)_{i \in [m]} = \mathbf{w}^\intercal \bmod q,$$

i.e. that the evaluation of $\mathsf{LDE}[\mathbf{w}]$ over the grid $[d]^\mu$ corresponds to the entries of $\mathbf{w}$. For any $\mathbf{r} = (r_i)_{i \in [\mu]} \in \mathcal{R}_q^\mu$, the evaluation $\mathsf{LDE}[\mathbf{w}](\mathbf{r})$ can be written as $\langle \widetilde{\mathbf{r}}, \mathbf{w} \rangle$ where

$$\widetilde{\mathbf{r}}^\intercal = \bigotimes_{j \in [\mu]} \left(\prod_{k' \in [d] \setminus \{k\}} \frac{r_j - k'}{k - k'} \right)_{k \in [d]} \bmod q$$

has a (row-)tensor structure. Therefore, relations of the form $\mathsf{LDE}[\mathbf{W}](\mathbf{r}) = s \bmod q$ with given $(\mathbf{r}, s)$ are readily captured by Ξ^{lin}.

We consider a tuple $((\mathbf{H}, \mathbf{F}, \mathbf{Y}, (\mathbf{r}_i, s_i)_{i \in [t]}), \mathbf{w})$ to be in the relation $\Xi^{\mathsf{lde}\text{-}\otimes}$ if (i) $((\mathbf{H}, \mathbf{F}, \mathbf{Y}), \mathbf{w}) \in \Xi^{\mathsf{lin}}$ and (ii) $\mathsf{LDE}[\mathbf{w}](\mathbf{r}_i) = s_i \bmod q$ for all $i \in [t]$. Since $\mathsf{LDE}[\mathbf{w}](\mathbf{r}_i) = s_i \bmod q \leftrightarrow \langle \widetilde{\mathbf{r}_i}, \mathbf{w} \rangle = s_i \bmod q$, LDE evaluation claims can be captured by Ξ^{lin}, therefore $\Xi^{\mathsf{lde}\text{-}\otimes}$ can naturally be reduced to Ξ^{lin} as described in Lemma 2.

Norm-Check. We propose a new norm-check protocol with a linear-time prover based on the sumcheck protocol to improve our framework [KLNO24, KLNO25], both of which had a quasi-linear time prover.

In this work, we measure the norm of elements by the ℓ_2-norm over the canonical embedding of $\mathcal{K}$. The norm of a vector $\mathbf{x} \in \mathcal{R}^m$ is denoted by $\|\mathbf{x}\|_{\sigma,p} \coloneqq \|\sigma(\mathbf{x})\|_2$. An important fact about such a norm is that

$$\begin{aligned} \|\mathbf{x}\|_{\sigma,2}^2 &= \sum_{i \in [\varphi]} \sum_{j \in [m]} |\sigma_i(x_j)|^2 = \sum_{i \in [\varphi]} \sum_{j \in [m]} \sigma_i(x_j) \cdot \overline{\sigma_i(x_j)} \\ &= \sum_{i \in [\varphi]} \sum_{j \in [m]} \sigma_i(x_j \overline{x_j}) = \mathsf{Trace}\left(\sum_{j \in [m]} x_j \overline{x_j} \right) = \mathsf{Trace}\left(\langle \mathbf{x}, \overline{\mathbf{x}} \rangle \right) \end{aligned}$$

where $\bar{\cdot}$ denotes the complex conjugate. We will use this fact to reduce a norm-check claim to an inner-product claim, which can be further reduced to a sumcheck claim. In more detail, we define a norm

[8] Also called face-splitting product.

relation Ξ^{norm} to be the same as Ξ^{lin} but with an additional input ν and the additional requirement that $\|\mathbf{W}\|_{\sigma,2} \leq \nu$.

Now, we define the function $f(x)$ as $f(x) = x \cdot \overline{x}$ for $x \in \mathcal{K}$. We observe that the "sumcheck claim" over $f(\mathsf{LDE}[\mathbf{w}])(\mathbf{x}) \coloneqq f(\mathsf{LDE}[\mathbf{w}](\mathbf{x}))$ precisely captures the inner-product $\langle \mathbf{w}, \overline{\mathbf{w}} \rangle$ as follows:

$$\langle \mathbf{w}, \overline{\mathbf{w}} \rangle = \sum_{j \in [m]} f(w_j) = \sum_{\mathbf{z} \in [d]^\mu} f(\mathsf{LDE}[\mathbf{w}])(\mathbf{z}) = t \bmod q.$$

Given that the witness is proved to be already of relatively small norm (via random projections as in [KLNO25]), the equation above also holds over $\mathcal{R}$ without the reduction modulo q. Therefore, the verifier can check the norm bound by checking $\mathsf{Trace}(t) \leq \nu^2$, effectively reducing the norm-check to a sumcheck claim.

The prover's next step is to prove that the sum t was computed correctly. To this end, we observe that $f(\mathsf{LDE}[\mathbf{w}])$ is a polynomial of degree $2(d-1)$ in each variable, and therefore by employing the sumcheck protocol, the sumcheck claim can be reduced to a single evaluation of $f(\mathsf{LDE}[\mathbf{w}])$ at a random point $\mathbf{r} \in \mathcal{R}_q^\mu$.

Then, the remaining task is to prove that the evaluation $f(\mathsf{LDE}[\mathbf{w}])(\mathbf{r}) = s \bmod q$ was computed correctly. For that, the prover sends s_0, s_1 so that

$$s_0 = \mathsf{LDE}[\mathbf{w}](\mathbf{r}) \bmod q \quad \text{and} \quad s_1 = \mathsf{LDE}[\overline{\mathbf{w}}](\mathbf{r}) \bmod q$$

and the verifier checks that $s = s_0 \cdot s_1 \bmod q$. By the conjugation identity $\overline{\mathsf{LDE}[\mathbf{w}](\mathbf{r})} = \mathsf{LDE}[\overline{\mathbf{w}}](\overline{\mathbf{r}}) \bmod q$, the claim $\mathsf{LDE}[\overline{\mathbf{w}}](\mathbf{r}) = s_1 \bmod q$ is equivalent to $\mathsf{LDE}[\mathbf{w}](\overline{\mathbf{r}}) = \overline{s_1} \bmod q$. Therefore the verifier's checks amount to two evaluation claims $(\mathbf{r}, s_0)$ and $(\overline{\mathbf{r}}, \overline{s_1})$, and we obtain the instance

$$((\mathbf{H}, \mathbf{F}, \mathbf{Y}, (\mathbf{r}, s_0), (\overline{\mathbf{r}}, \overline{s}_1)), \mathbf{w}) \in \Xi^{\mathsf{lde}\text{-}\otimes}.$$

R1CS, Improved batching and Sumcheck Relation. Our norm-check described above is only one of the new RoKs we develop in this work and is itself a composition of several RoKs. The most notable one is the sumcheck RoK, which we provide in multiple variants to capture different types of sumcheck claims.

From the sumcheck RoKs we construct:

(i) A RoK for the R1CS relation, which is a natural extension of the R1CS relation over fields to ring settings, defined similarly to the one in [BC25a]. The RoK follows the linearisation strategy of [KS24, BC25a] and reduces the R1CS relation to evaluation claims over the LDE of the witness and matrices that define the R1CS instance. These are reduced to $\Xi^{\mathsf{lde}\text{-}\otimes}$ as described above.

(ii) A RoK for improved batching of structured linear relations, which is a more elegant variant of the batching RoK in [KLNO25] by employing the sumcheck protocol. In [KLNO25], the batching is done by randomly combining multiple structured linear relations into a single one via the matrix $\mathbf{H}$. This approach only reduces the communication complexity but not the prover time. The sumcheck-based batching RoK works by expressing the bottom rows $\underline{\mathbf{F}}\mathbf{w} = \underline{\mathbf{y}} \bmod q$ of $\mathbf{F}\mathbf{w} = \mathbf{y} \bmod q$ (or the rows to be compressed) as a set of sumcheck claims:

$$\sum_{j \in [m]} \mathsf{LDE}[\underline{\mathbf{f}}_i](\mathbf{z}) \cdot \mathsf{LDE}[\mathbf{w}](\mathbf{z}) = \underline{y}_i \bmod q,$$

where $\underline{\mathbf{f}}_i$ is the i-th row of $\underline{\mathbf{F}}$ and $\underline{y}_i$ is the i-th entry of $\underline{\mathbf{y}}$. The sumcheck claims are batched and then reduced to a single evaluation claim in $\Xi^{\mathsf{lde}\text{-}\otimes}$ as described above. Since this RoK actually reduces the number of rows of $\mathbf{F}$, the compressing matrix $\mathbf{H}$ is no longer needed and can be set to an identity matrix.

1.4 Paper Organisation

This paper is organised as follows: (i) In Section 2, we provide the necessary background on algebraic number theory and the principal linear relation Ξ^{lin} considered in this work as well as in RPS/RnR. (ii) In Section 3, we introduce the toolbox involving sumcheck techniques, which are used to construct the norm-check protocol in Section 4. Specifically, we formalise the low-degree extension relations $\Xi^{\mathsf{lde}(\text{-}\otimes)}$

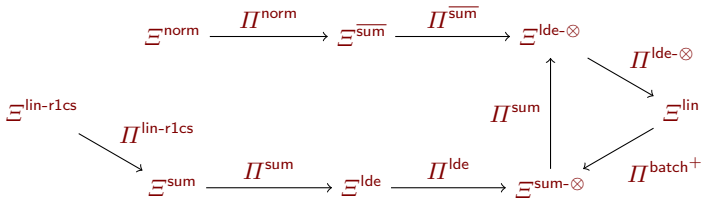

Fig. 1: Overview of results.

along with the reduction from $\Xi^{\mathsf{lde}\text{-}\otimes}$ to Ξ^{lin}, a sumcheck relation $\Xi^{\overline{\mathsf{sum}}}$, and the RoK connecting these relations, $\Pi^{\overline{\mathsf{sum}}} : \Xi^{\overline{\mathsf{sum}}} \to \Xi^{\mathsf{lde}\text{-}\otimes}$. (iii) Section 4 presents the construction of our norm-check protocol, along with its correctness and security proofs. (iv) In Section 5, we demonstrate three applications instantiated by our framework, namely, a SNARK, a PCS, and a folding scheme. The experimental results for each application are described in their respective sections, while exhaustive implementation details are deferred to Appendix E. Formal reductions specific to the folding scheme application are deferred to Appendix D. (v) In Section 7, we define the committed R1CS relation $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ and provide Lemma 5, which establishes the RoK $\Pi^{\mathsf{lin\text{-}r1cs}}$ from $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ to the principal relation Ξ^{lin}. The detailed description of the formal reductions and their building blocks is deferred to Appendices B and C.

The overview of the proposed relations and RoKs is illustrated in Fig. 1.

2 Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$ denote natural numbers and $\lambda \in \mathbb{N}$ be the security parameter. For $m, n \in \mathbb{N}$, we write $[n] \coloneqq \{0, \ldots, n-1\}$ counting from 0 and $[m : n] \coloneqq [n] \setminus [m]$. For multidimensional ranges, we use the shorthand $(i, j, k) \in [n, m, \ell]$ for $i \in [n], j \in [m]$, and $k \in [\ell]$. To represent $\mathbb{Z}_q$ we use the balanced representation, i.e. $\{-\lceil q/2 \rceil + 1, \ldots, \lfloor q/2 \rfloor\}$. We use bold lower-case letters to denote vectors, e.g. $\mathbf{v}$, and bold upper-case letters to denote matrices, e.g. $\mathbf{M}$. For matrices (or vectors) $\mathbf{M}_0, \ldots, \mathbf{M}_{k-1}$ of appropriate dimensions, we write $(\mathbf{M}_i)_{i \in [k]}$ for horizontal concatenation. We write $\mathbf{0}_r$ (resp., $\mathbf{1}_r$) for the r-dimensional vector with all entries set to 0 (resp., 1). We denote by $\mathbf{e}_{i,r}$ a vector which is equal to $\mathbf{0}_r$ except at the i-th position, which is set to 1. We omit r and write simply $\mathbf{e}_i$ if r is implied from the context. This notation naturally generalises to matrices, i.e. $\mathbf{E}_{i,j,m,n}$ simplifies to $\mathbf{E}_{i,j}$

2.1 Algebraic Number Theory

Cyclotomic Fields. Throughout this work, we let $\mathcal{K} = \mathbb{Q}(\zeta)$ be a cyclotomic field with conductor $\mathfrak{f}$ of degree $\varphi = \varphi(\mathfrak{f})$, where ζ is a root of unity of order $\mathfrak{f}$ and φ is Euler's totient function, and $\mathcal{O}_{\mathcal{K}} = \mathbb{Z}[\zeta]$ be its ring of integers. For a modulus $q \in \mathbb{N}$, we write $\mathcal{R}_q \coloneqq \mathcal{R}/q\mathcal{R}$. We always assume that q is a prime number. We denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ the sets of units in $\mathcal{R}$ and $\mathcal{R}_q$ respectively. We endow $\mathcal{R}$ with a geometry via the canonical embedding $\sigma : \mathcal{K} \to \mathbb{C}^\varphi$. Specifically, for a given $\mathbb{Z}$-basis $\mathbf{b} = (b_i)_{i \in [\varphi]}$ of $\mathcal{R}$ and an element $x = \sum_{i \in [\varphi]} x_i b_i \in \mathcal{R}$, we write $\mathfrak{cf}_{\mathbf{b}}(x) \coloneqq (x_i)_{i \in [\varphi]}$ and $\sigma(x) \coloneqq (\sigma_j(x))_{j \in [\varphi]}$, where $\sigma_j \in \mathsf{Gal}(\mathcal{K}/\mathbb{Q})$. We extend the notation of σ naturally to vectors, i.e. if $\mathbf{x} = (x_i)_{i \in [m]} \in \mathcal{R}^m$, then $\sigma(\mathbf{x}) \coloneqq (\sigma(x_i))_{i \in [m]}$ is defined as concatenations.

Norm and Trace. The ℓ_p-norm of a vector $\mathbf{x} \in \mathcal{R}^m$ is denoted by $\|\mathbf{x}\|_{\sigma,p} \coloneqq \|\sigma(\mathbf{x})\|_p$. We will mostly use $\|\cdot\|_{\sigma,2}$. For a matrix $\mathbf{M} \in \mathcal{R}^{n \times m}$, the norm is defined as $\|\mathbf{M}\|_{\sigma,p} = \max_{i \in [n]} \|\mathbf{m}_i\|_{\sigma,p}$, where $\mathbf{m}_i$ is the i-th column of the matrix. Note that $\|\mathbf{x}\|_{\sigma,2}^2 = \mathsf{Trace}(\langle \mathbf{x}, \overline{\mathbf{x}} \rangle)$, where $\bar{\cdot}$ denotes the complex conjugate. For a Galois extension $\mathcal{M}/\mathcal{L}$, the field trace can be computed as $\mathsf{Trace}_{\mathcal{M}/\mathcal{L}} : \mathcal{K} \to \mathcal{L}$, $\mathsf{Trace}_{\mathcal{M}/\mathcal{L}}(x) \coloneqq \sum_{\sigma_j \in \mathsf{Gal}(\mathcal{K}/\mathcal{L})} \sigma_j(x)$. When $\mathcal{L} = \mathbb{Q}$, we write $\mathsf{Trace} = \mathsf{Trace}_{\mathcal{M}/\mathbb{Q}}$.

Ring Splitting. When q has multiplicative order e modulo $\mathfrak{f}$ then $\mathcal{R}_q$ splits into φ/e finite fields, i.e. $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}$. We denote this isomorphism as $\mathsf{CRT} : \mathcal{R}_q \to (\mathbb{F}_{q^e})^{\varphi/e}$ and its inverse as $\mathsf{CRT}^{-1} : (\mathbb{F}_{q^e})^{\varphi/e} \to \mathcal{R}_q$. We naturally generalise this notation to vectors, i.e. $\mathsf{CRT}(\mathbf{x}) : \mathcal{R}_q^m \to (\mathbb{F}_{q^e})^{m\varphi/e}$ is defined as the concatenation of CRT applied to each entry of the vector. Furthermore, we define CRT also for polynomials, i.e. $\mathsf{CRT} : \mathcal{R}_q^r[x^\mu] \to \mathbb{F}_{q^e}^{r\varphi/e}[x^\mu]$ applies CRT to each coefficient vector of the polynomial.

2.2 RoK, Paper, SISsors [KLNO24]

Principal relation. We recall the principal relation handled by most of the reductions of knowledge in [KLNO24].

$$\Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta} := \left\{ \begin{array}{l} ((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}) \\ \quad \mathbf{H} \in \mathcal{R}_q^{\hat{n}\times n} \quad \mathbf{F} \in \mathcal{R}_q^{n\times m} \\ \quad \mathbf{Y} \in \mathcal{R}_q^{\hat{n}\times r} \quad \mathbf{W} \in \mathcal{R}^{m\times r} \\ \quad \|\mathbf{W}\|_{\sigma,2} \le \beta \ \text{ and } \ \mathbf{HFW} = \mathbf{Y} \bmod q \end{array} \right\}$$

where $\mathbf{H}$ is restricted to take the form $\mathbf{H} = \begin{pmatrix} \mathbf{I}_{\overline{n}} & \\ & \underline{\mathbf{H}} \end{pmatrix}$ for some $\overline{n} \le n$. Throughout, we think of $n = \overline{n} + \underline{n}$ and $\mathbf{F} = \begin{pmatrix} \overline{\mathbf{F}} \\ \underline{\mathbf{F}} \end{pmatrix} \in \mathcal{R}_q^{(\overline{n}+\underline{n})\times m}$ as consisting of a top part $\overline{\mathbf{F}} \in \mathcal{R}_q^{\overline{n}\times m}$ and a bottom part $\underline{\mathbf{F}} \in \mathcal{R}_q^{\underline{n}\times m}$. We assume that $\mathbf{F}$ can be decomposed into a row-tensor $\mathbf{F} = \mathbf{F}_0 \bullet \ldots \mathbf{F}_{\mu-1}$ where $\mathbf{F} \in \mathcal{R}_q^{n\times d}$, shorthanded as $\mathbf{F} \in \mathcal{R}_q^{n\times d^{\otimes\mu}}$.

SIS Break. To account for scenarios where a prover can solve a (vanishing) SIS problem, we define the relation Ξ^{sis}, whose statements share the same space as Ξ^{lin}. This allows us to conveniently express $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{sis}}$:

$$\Xi^{\mathsf{sis}}_{\mathcal{R},q,m,\hat{n},r,\beta} := \left\{ \begin{array}{l} ((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{x}): \\ \quad \mathbf{H} \in \mathcal{R}_q^{\hat{n}\times n} \quad \mathbf{F} \in \mathcal{R}_q^{n\times m} \\ \quad \mathbf{Y} \in \mathcal{R}_q^{\hat{n}\times r} \quad \mathbf{W} \in \mathcal{R}^{m\times r} \\ \quad \|\mathbf{x}\|_{\sigma,2} \le \beta \ \text{ and } \ \overline{\mathbf{F}}\mathbf{x} = \mathbf{0} \bmod q \end{array} \right\}$$

where $\mathbf{H}$ is as above. To enable extractors to handle SIS-breaks, we formally require knowledge reduction claims of the form $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{sis}} \leftarrow \Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{sis}}$ (cf. [KLNO24]). For simplicity, we reduce this to $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{sis}} \leftarrow \Xi^{\mathsf{lin}}$ or even $\Xi^{\mathsf{lin}} \leftarrow \Xi^{\mathsf{lin}}$, when managing SIS-breaks is straightforward.

3 Ξ^{lde}, $\Xi^{\mathsf{lde}\text{-}\otimes}$ and $\Xi^{\overline{\mathsf{sum}}}$: LDE and Sumcheck Relations

In this section, we introduce two key families of relations that form the foundation for the remainder of this work: the low-degree extension relations, Ξ^{lde} and $\Xi^{\mathsf{lde}\text{-}\otimes}$, and the sumcheck relation $\Xi^{\overline{\mathsf{sum}}}$. Both are based on the perspective of treating the witness $\mathbf{W}$ as a low-degree polynomial extension.

(i) In Section 3.1 we define the low-degree extension (LDE) relations Ξ^{lde} and $\Xi^{\mathsf{lde}\text{-}\otimes}$ and present a reduction of $\Xi^{\mathsf{lde}\text{-}\otimes}$ to Ξ^{lin}. (ii) In Section 3.2 we define the sumcheck relation $\Xi^{\overline{\mathsf{sum}}}$, which is defined as a sumcheck relation over the low-degree extension relations Ξ^{lde} and $\Xi^{\mathsf{lde}\text{-}\otimes}$ respectively.

3.1 $\Pi^{\mathsf{lde}\text{-}\otimes}$: RoK from $\Xi^{\mathsf{lde}\text{-}\otimes}$ to Ξ^{lin}

Definition 1 (Low-Degree Extension (LDE)). *Let $\mathbf{w}^\intercal = (w_{\mathbf{z}})_{\mathbf{z}\in[d]^\mu} \in \mathcal{K}^{1\times d^\mu}$. A degree-$d$ extension of $\mathbf{w}$ is a μ-variate polynomial $\mathsf{LDE}_d[\mathbf{w}] : \mathcal{K}^\mu \to \mathcal{K}$ with individual degree d satisfying $\forall\ \mathbf{z} \in [d]^\mu,\ \mathsf{LDE}_d[\mathbf{w}](\mathbf{z}) = w_{\mathbf{z}}$. For a matrix $\mathbf{W} \in \mathcal{K}^{d^\mu\times r}$, $\mathsf{LDE}_d[\mathbf{W}]$ is defined as the column-wise concatenation of $(\mathsf{LDE}_d[\mathbf{w}_j])_{j\in[r]}$. In other words, $\mathsf{LDE}_d[\mathbf{W}]$ is a μ-variate polynomial $\mathsf{LDE}_d[\mathbf{w}] : \mathcal{K}^\mu \to \mathcal{K}^r$ with individual degree $d-1$ satisfying $\forall\ \mathbf{z} \in [d]^\mu,\ \mathsf{LDE}[\mathbf{W}](\mathbf{z}) = (\mathsf{LDE}[w_{\mathbf{z},j}](\mathbf{z}))_{j\in[r]} = \mathbf{w}_{\mathbf{z}}^\intercal$ where $\mathbf{w}_{\mathbf{z}}$ is the row of $\mathbf{W}$ indexed by $\mathbf{z}$.*

We will omit the subscript d as it is fixed throughout this work. The following lemma follows immediately from the fact that a univariate degree-d polynomial over a field can have at most d distinct roots and by induction over $j \in [\mu]$.

Lemma 1. *For any $\mathbf{W} \in \mathcal{K}^{d^\mu\times r}$, $\mathsf{LDE}[\mathbf{W}]$ is uniquely determined by $\forall\ \mathbf{x} \in \mathcal{K}^\mu,\ \mathsf{LDE}[\mathbf{W}](\mathbf{x})^\intercal = \widetilde{\mathbf{x}}^\intercal\mathbf{W}$, where $\widetilde{\mathbf{x}}^\intercal = \bigotimes_{j\in[\mu]} \left(\prod_{k'\in[d]\setminus\{k\}} (x_j - k')/(k - k')\right)_{k\in[d]}$.*

We define the LDE relation Ξ^{lde}, which is almost identical to Ξ^{lin}, except that a statement further includes tuples $(\mathbf{r}_i, \mathbf{s}_i, \mathbf{M}_i)_i$ and the witness is further required to satisfy $\mathsf{LDE}[\mathbf{M}_i\mathbf{W}](\mathbf{r}_i) = \mathbf{s}_i \bmod q$ for all i. For structured $\mathbf{M}_i \in \mathcal{R}_q^{d^{\otimes\tilde{\mu}} \times d^{\otimes\mu}}$, it is clear that $\mathsf{LDE}[\mathbf{M}_i\mathbf{W}](\mathbf{r}_i)^{\mathsf{T}}$ can be written as $\widetilde{\mathbf{r}}_i^{\mathsf{T}} \cdot \mathbf{M}_i \cdot \mathbf{W}$ where

$$\begin{aligned}\widetilde{\mathbf{r}}_i^{\mathsf{T}} \cdot \mathbf{M}_i &= \left(\bigotimes_{j\in[\tilde{\mu}]} \left(\prod_{k'\in[d]\setminus\{k\}} (r_{i,j} - k')/(k-k')\right)_{k\in[d]}\right) \cdot \left(\bigotimes_{j\in[\tilde{\mu}]} \mathbf{M}_{i,j}\right) \\ &= \bigotimes_{j\in[\tilde{\mu}]} \left(\left(\prod_{k'\in[d]\setminus\{k\}} (r_{i,j} - k')/(k-k')\right)_{k\in[d]} \cdot \mathbf{M}_{i,j}\right) \bmod q\end{aligned}$$

holds by the mixed-product property of the Kronecker product and still has a (row-)tensor structure. In other words, $\Xi^{\mathsf{lde}\text{-}\otimes}$ reduces trivially to Ξ^{lin}.

Definition 2 (Ξ^{lde} and $\Xi^{\mathsf{lde}\text{-}\otimes}$: LDE Relations). *The relation Ξ^{lde} is*

$$\Xi^{\mathsf{lde}(\text{-}\otimes)}_{\hat{n},n,\mu,\tilde{\mu},r,\beta,t} := \left\{ \begin{array}{l} ((\mathbf{H},\mathbf{F},\mathbf{Y},(\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]}),\mathbf{W}): \\ \quad ((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}) \in \Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta} \\ \quad (\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]} \in (\mathcal{R}_q^{\tilde{\mu}} \times \mathcal{R}_q^{r} \times \mathcal{R}_q^{\tilde{m}\times m})^t \\ \quad \forall_{i\in[t]} \;\; \mathbf{s}_i = \mathsf{LDE}[\mathbf{M}_i\mathbf{W}](\mathbf{r}_i) \bmod q \end{array} \right\}.$$

The structured variant $\Xi^{\mathsf{lde}\text{-}\otimes}$ is identical to Ξ^{lde}, except that the matrices $\mathbf{M}_i$ in the statement are required to be in $\mathcal{R}_q^{d^{\otimes\tilde{\mu}} \times d^{\otimes\mu}}$. If $\mathbf{M}_i = \mathbf{I}_m$ (hence structured), then it is omitted from the statement.

The following lemma follows immediately from the observation that, if $\mathbf{M} \in \mathcal{R}_q^{d^{\otimes\tilde{\mu}} \times d^{\otimes\mu}}$ and $\widetilde{\mathbf{r}}$ is as defined in Lemma 1, then $\widetilde{\mathbf{r}}^{\mathsf{T}} \cdot \mathbf{M}$ can be written as a tensor product of μ vectors each of d dimension.

Lemma 2. *Let $\Pi^{\mathsf{lde}\text{-}\otimes}$ be the protocol which on input a $\Xi^{\mathsf{lde}\text{-}\otimes}_{\hat{n},n,\mu,\tilde{\mu},r,\beta,t}$ instance $((\mathbf{H},\mathbf{F},\mathbf{Y},(\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]}),\mathbf{W})$ outputs a $\Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta}$ instance $((\mathbf{H}',\mathbf{F}',\mathbf{Y}'),\mathbf{W})$ where*

$$\mathbf{H}' := \begin{bmatrix} \mathbf{H} & \\ & \mathbf{I}_t \end{bmatrix} \qquad \mathbf{F}' := \begin{bmatrix} \mathbf{F} \\ \left((\mathbf{M}_i^{\mathsf{T}}\widetilde{\mathbf{r}}_i)_{i\in[t]}\right)^{\mathsf{T}} \end{bmatrix} \qquad \mathbf{Y}' := \begin{bmatrix} \mathbf{Y} \\ \left((\mathbf{s}_i)_{i\in[t]}\right)^{\mathsf{T}} \end{bmatrix}$$

and $(\widetilde{\mathbf{r}}_i)_{i\in[t]}$ are defined as in Lemma 1. The protocol $\Pi^{\mathsf{lde}\text{-}\otimes}$ is a perfectly correct and knowledge sound reduction of knowledge for:

$$\Xi^{\mathsf{lde}\text{-}\otimes}_{\hat{n},n,\mu,\tilde{\mu},r,\beta,t} \leftrightarrow \Xi^{\mathsf{lin}}_{\hat{n}+t,n+t,\mu,r,\beta}.$$

The communication cost is 0.

3.2 $\Pi^{\overline{\mathsf{sum}}}$: RoK from $\Xi^{\overline{\mathsf{sum}}}$ to $\Xi^{\mathsf{lde}\text{-}\otimes}$

The reduction $\Pi^{\overline{\mathsf{sum}}}$ is a sumcheck-based protocol which reduces a $\Xi^{\overline{\mathsf{sum}}}$ instance to a $\Xi^{\mathsf{lde}\text{-}\otimes}$ instance. $\Xi^{\mathsf{lde}\text{-}\otimes}$ is defined in Definition 3. The protocol is shown in Fig. 2 and a formal lemma is given in Lemma 3.

Definition 3 ($\Xi^{\overline{\mathsf{sum}}}$: Sumcheck Relation). *The relation $\Xi^{\overline{\mathsf{sum}}}$ is defined as*

$$\Xi^{\overline{\mathsf{sum}}}_{\hat{n},n,\mu,r,\beta} := \left\{ \begin{array}{l} ((\mathbf{H},\mathbf{F},\mathbf{Y},\mathbf{t}),\mathbf{W}): \\ \quad ((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}) \in \Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta} \\ \quad \sum_{\mathbf{z}\in[d]^\mu} (\mathsf{LDE}[\mathbf{W}] \odot \mathsf{LDE}[\overline{\mathbf{W}}])(\mathbf{z}) = \mathbf{t} \bmod q \in \mathcal{R}_q^r \end{array} \right\}.$$

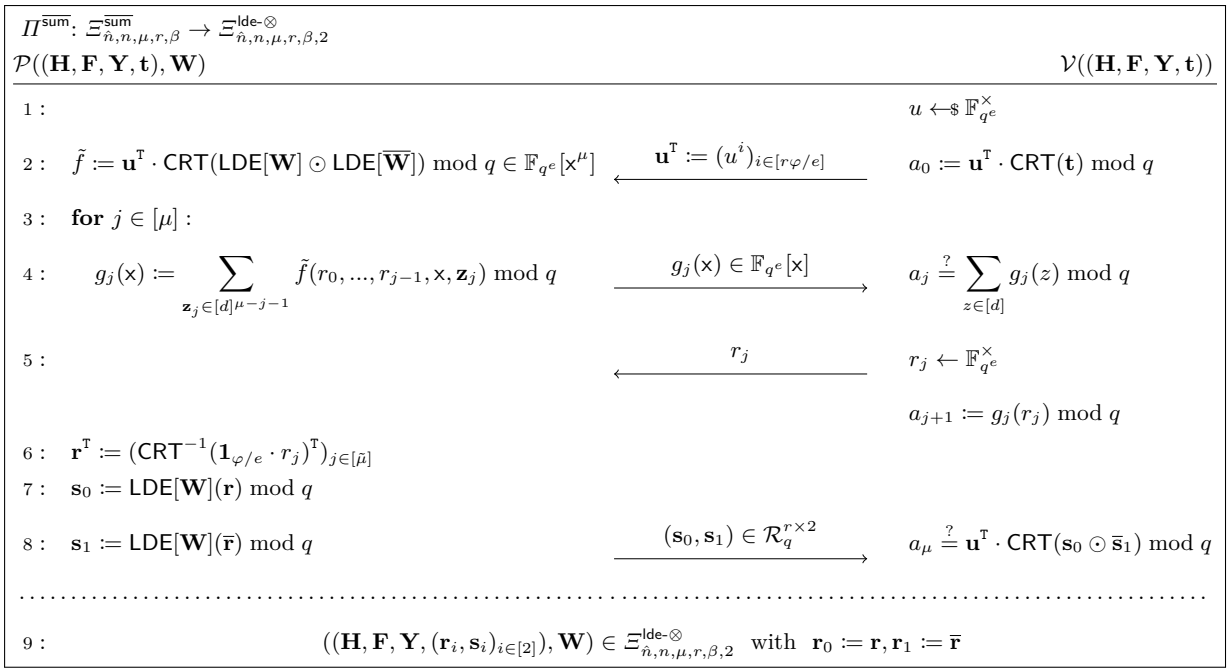

Fig. 2: $\Pi^{\overline{\mathsf{sum}}}$: RoK from $\Xi^{\overline{\mathsf{sum}}}$ to $\Xi^{\mathsf{lde}\text{-}\otimes}$.

Lemma 3. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, \mu \in \mathbb{N}$ and $0 \le \beta \le \beta' \le q$. $\Pi^{\overline{\mathsf{sum}}}$ is a perfectly correct reduction of knowledge for:*

$$\Xi^{\overline{\mathsf{sum}}}_{\hat{n},n,\mu,r,\beta} \to \Xi^{\mathsf{lde}\text{-}\otimes}_{\hat{n},n,\mu,r,\beta,2}$$

It is knowledge sound with knowledge-error $\kappa = \frac{2\mu(d-1)+r\varphi/e-1}{q^e}$ for

$$\Xi^{\overline{\mathsf{sum}}}_{\hat{n},n,\mu,r,\beta'} \cup \Xi^{\mathsf{vsis}}_{2\beta'} \leftarrow \Xi^{\mathsf{lde}\text{-}\otimes}_{\hat{n},n,\mu,r,\beta',2}$$

The communication cost is $(2d-1)\mu e \log q + 2r \log|\mathcal{R}_q|$. Prover runs in $\mathcal{O}(rm)$ ring operations.

Proof. Correctness: For correctness, we observe that

$$\begin{aligned} a_j = g_{j-1}(r_{j-1}) &= \sum_{\mathbf{z}_{j-1}\in[d]^{\mu-j}} \tilde{f}(r_0,...,r_{j-1},\mathbf{z}_{j-1}) \\ &= \sum_{z\in[d]} \sum_{\mathbf{z}_j\in[d]^{\mu-j-1}} \tilde{f}(r_0,...,r_{j-1},z,\mathbf{z}_j) = \sum_{z\in[d]} g_j(z) \bmod q \end{aligned}$$

Further, $\tilde{f}(r_0,\ldots,r_{\mu-1}) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(f \odot \bar{f}(\mathbf{r})) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(f(\mathbf{r}) \odot \overline{f(\bar{\mathbf{r}})}) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{s}_0 \odot \overline{\mathbf{s}_1}) \bmod q$ for $f \coloneqq \mathsf{LDE}[\mathbf{W}]$ and $\bar{f} \coloneqq \mathsf{LDE}[\overline{\mathbf{W}}]$.

Knowledge Soundness: Without loss of generality, suppose $\mathcal{P}^*$ is deterministic and succeeds with a probability ϵ. $\mathcal{E}$ runs $\mathcal{P}^*$ on a random challenge $(u,\mathbf{r})$. If the prover fails, $\mathcal{E}$ aborts. On success, $\mathcal{E}$ obtains $(\mathbf{W},(g_j)_{j\in[\mu]},\mathbf{s}_0,\mathbf{s}_1)$ satisfying:

$$\begin{aligned} &((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}) \in \Xi^{\mathsf{lin}}, \quad \|\mathbf{W}\|_{\sigma,2} \le \beta', \\ &f(\mathbf{r}) = \mathbf{s}_0 \bmod q, \ \ f(\bar{\mathbf{r}}) = \mathbf{s}_1 \bmod q, \quad \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{t}) = \sum_{z\in[d]} g_0(z) \bmod q, \\ &g_j(r_j) = \sum_{z\in[d]} g_{j+1}(z) \ \ \forall\ j \in [\mu-1], \quad g_{\mu-1}(r_{\mu-1}) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{s}_0 \odot \bar{\mathbf{s}}_1) \bmod q. \end{aligned}$$

If $\mathbf{W}$ satisfies $\Xi^{\overline{\mathsf{sum}}}$, the extractor outputs $\mathbf{W}$ and terminates. Otherwise, it re-runs $\mathcal{P}^*$ on fresh challenges $(u',\mathbf{r}')$ until it obtains a second accepting transcript $(\mathbf{W}',(g'_j)_{j\in[\mu]},\mathbf{s}'_0,\mathbf{s}'_1)$ satisfying

$$((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}') \in \Xi^{\mathsf{lin}}, \quad \|\mathbf{W}'\|_{\sigma,2} \le \beta',$$

$$f(\mathbf{r}') = \mathbf{s}'_0 \bmod q,\ f(\overline{\mathbf{r}'}) = \mathbf{s}'_1 \bmod q, \quad \mathbf{u'}^{\mathsf{T}}\mathsf{CRT}(\mathbf{t}) = \sum_{z\in[d]} g'_0(z) \bmod q,$$

$$g'_j(r'_j) = \sum_{z\in[d]} g'_{j+1}(z) \ \ \forall\ j \in [\mu-1], \quad a'_\mu = g'_{\mu-1}(r'_{\mu-1}) = \mathbf{u'}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{s'}_0 \odot \bar{\mathbf{s}}'_1) \bmod q.$$

If $\mathbf{W}' \neq \mathbf{W} \bmod q$, then any non-zero column of $\mathbf{W}' - \mathbf{W}$ is a solution $\mathbf{v}$ to $\overline{\mathbf{F}}\mathbf{v} = \mathbf{0} \bmod q$, and hence $(\overline{\mathbf{F}}, \mathbf{v}) \in \Xi^{\mathsf{vsis}}_{2\beta'}$.

Knowledge error: As we suppose that $((\mathbf{H}, \mathbf{F}, \mathbf{Y}, \mathbf{t}), \mathbf{W}) \notin \Xi^{\overline{\mathsf{sum}}}$, we show that, conditioned on the conditions above, this happens with κ/ϵ probability.

We have that since $f(\mathbf{r}') = \mathbf{s}'_0 \bmod q$ and $f(\bar{\mathbf{r}}') = \mathbf{s}'_1 \bmod q$, then $\tilde{f}(\mathbf{r}') = a'_\mu \bmod q$. Now, the proof is by induction on μ. For $\mu = 1$, observe that the prover's only message specifies a degree $2(d-1)$ univariate polynomial $g'_0(\mathsf{x})$. If $g'_0(\mathsf{x}) \neq \tilde{f}(\mathsf{x}) \bmod q$, then because any two distinct degree $2(d-1)$ univariate polynomials over finite field $\mathbb{F}_{q^e}$ can agree at most $2(d-1)$ inputs, $g'_0(r'_0) = \tilde{f}(r'_0) \bmod q$ with probability at most $2(d-1)/(\epsilon q^e)$ over the choice of r'_0. By induction and union bound, for all μ univariate polynomials, the sumcheck protocol has knowledge soundness error at most κ_0/ϵ, where $\kappa_0 \coloneqq 2\mu(d-1)/q^e$.

As established, unless with a κ_0/ϵ probability,

$$\sum_{\mathbf{z}\in[d]^{\tilde{\mu}}} \tilde{f}(\mathbf{z}) = a'_\mu \bmod q \tag{1}$$

and

$$\mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathsf{LDE}[\mathbf{W}] \odot \mathsf{LDE}[\overline{\mathbf{W}}]) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{s}_0 \odot \bar{\mathbf{s}}_1).$$

Therefore, if $\mathsf{LDE}[\mathbf{W}] \odot \mathsf{LDE}[\overline{\mathbf{W}}] \neq \mathbf{s}_0 \odot \bar{\mathbf{s}}_1$, then Eq. (1) fails unless with probability at most κ_1/ϵ, where $\kappa_1 \coloneqq (r\varphi/e-1)/q^e$ by the Schwartz-Zippel lemma. Otherwise, we yield a contradiction with a supposition that $((\mathbf{H}, \mathbf{F}, \mathbf{Y}, \mathbf{t}), \mathbf{W}) \notin \Xi^{\overline{\mathsf{sum}}}$.

We conclude that $\mathcal{E}$ outputs either (i) $\mathbf{W}$ satisfying the $\Xi^{\overline{\mathsf{sum}}}$ relation, or (ii) $\mathbf{v}$ satisfying $(\overline{\mathbf{F}}, \mathbf{v}) \in \Xi^{\mathsf{vsis}}_{2\beta'}$ with a probability

$$\epsilon(1 - \frac{\kappa_0 + \kappa_1}{\epsilon}) \leq \epsilon - \frac{2\mu(d-1) + r\varphi/e - 1}{q^e} = \epsilon - \kappa.$$

Expected runtime of $\mathcal{E}$: Let $\epsilon > 0$ be the success probability of $\mathcal{P}^*$ on a random challenge $(u, \mathbf{r})$. The extractor strategy is: (i) Run $\mathcal{P}^*$ on a fresh challenge; if unsuccessful, abort. Otherwise, record the successful transcript. (ii) Rewind and repeat with fresh challenges until a second accepting transcript. The number of prover invocations follows $\mathbb{E}[T] = (1-\epsilon) + \epsilon \cdot \left(1 + \frac{1}{\epsilon}\right) = 2$, so the extractor runs in expected polynomial time.

Prover Runtime: To analyse the complexity, we observe that the cost of operations "outside the loop" is naturally bounded by $\mathcal{O}(mr)$.

Inside the loop, the naive instantiation is insufficient as it runs in time $\mathcal{O}(\mu\tilde{m})$. The runtime of Π^{sum} is optimised via dynamic programming techniques from [Tha13, XZZ+19, CBBZ23]. The "dynamic programming optimisation" could be abstracted out as a way to compute univariate polynomials $g_i(\mathsf{x})$ by storing partially evaluated polynomials from the previous round.

- At the beginning $a_0 \coloneqq \sum_{\mathbf{z}\in[d]^\mu} f(\mathbf{z}) \bmod q$ is evaluated and we store all of the intermediate evaluations, $\tilde{f}_{0,i} \coloneqq \sum_{\mathbf{z}_i\in[d]^{\mu-i-1}} \tilde{f}(\mathsf{x}_0, \ldots \mathsf{x}_i, \mathbf{z}) \bmod q \ \ \forall\ i \in [\mu]$.
- We observe that $g_j = \tilde{f}_{j,j} \bmod q$ and therefore is sent without any computation. After receiving r_j in the j-th round, for all $i > j$, $\tilde{f}_{j+1,i} \bmod q$ is updated and $\tilde{f}_{j+1,i} \coloneqq \tilde{f}_{j,i}(r_j, \mathsf{x}_{j+1}, \ldots \mathsf{x}_i) \bmod q$.

For the runtime, we observe that the pre-computation phase requires linear (in the number of coefficients $\tilde{m}$ and by counting the operations over $\mathcal{R}_q$) time. Also, the first round requires the partial computation of polynomials of total degrees $2m/d - 1$, $2m/d^2 - 1$ down to $2m - 1$ via the geometric series. Therefore, it also requires a linear time, i.e. $\mathcal{O}(m)$. Each consecutive round requires a computation "cheaper" by a factor of d. As a result, again employing a geometric series, the sum of the runtimes spent on each round aggregates to a linear time, i.e. $\mathcal{O}(m)$.

Communication cost: We observe that communication includes sending degree $2(d-1)$ polynomials $(g_i)_{i\in[\mu]}$ represented by $2(d-1)+1$ coefficients, and 2 vectors of length r. In total, the communication cost is $(2d-1)\mu e \log q + 2r \log |\mathcal{R}_q|$. □

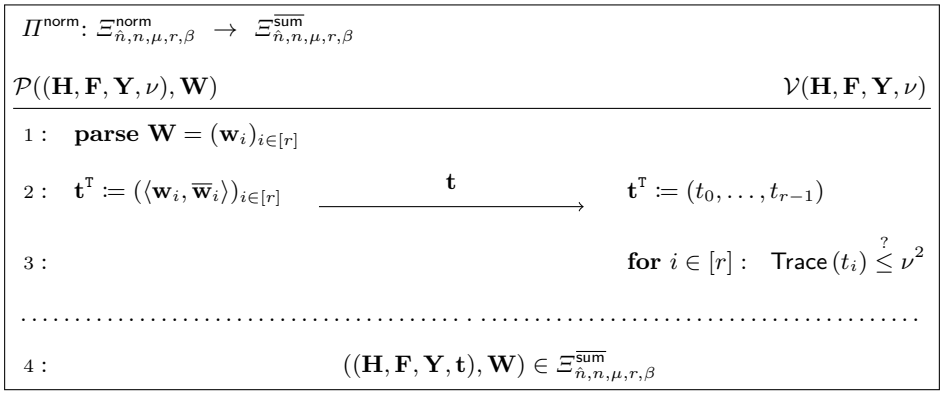

Fig. 3: Π^{norm}: a RoK from Ξ^{norm} to $\Xi^{\overline{\mathsf{sum}}}$.

Remark 1. We note the the presented $\Pi^{\overline{\mathsf{sum}}}$ protocol could be simply generalised to support complete family of automorphisms $\mathsf{Aut} := \{\sigma : \zeta \mapsto \zeta^k \mid k \in (\mathbb{Z}/\mathfrak{f}\mathbb{Z})^\times\}$ so that the sumcheck claim becomes

$$\sum_{\mathbf{z}\in[d]^{\bar{\mu}}} \bigodot_{\sigma\in S} \mathsf{LDE}[\sigma(\mathbf{W})](\mathbf{z}) = \mathbf{t} \bmod q.$$

where $S \subseteq \mathsf{Aut}$ is some subset of automorphisms. The alterations follows directly from the observation that

$$\sigma^{-1}(\mathsf{LDE}[\sigma(\mathbf{W})](\mathbf{r})) = \mathsf{LDE}[\mathbf{W}](\sigma^{-1}(\mathbf{r})) \bmod q.$$

so the prover sets $\mathbf{s}_i \coloneqq \mathsf{LDE}[\mathbf{W}](\sigma_i^{-1}(\mathbf{r})) \bmod q$ for $\sigma_i \in S$. The verifier checks the claim

$$a_\mu \overset{?}{=} \mathbf{u}^\intercal \cdot \mathsf{CRT}\left(\bigodot_{(i,\sigma)\in[|S|,S]} \sigma(\mathbf{s}_i)\right) \bmod q.$$

The knowledge extractor works analogously and the soundesss error becomes $\kappa = \frac{(|S|\mu(d-1)+r\varphi/e-1)}{q^e}$. The communication cost is $(|S|d-1)\mu e \log q + |S| r \log|\mathcal{R}_q|$. The growth in the knowledge error and communication cost is due to the higher degree of the sumcheck polynomial.

4 Norm-Check based on Sumcheck RoK

This section focuses on designing the RoK Π^{norm} for the relation Ξ^{norm}. In [KLNO24] a RoK $\Pi^{\text{norm-klno24}}$ is given which reduces a Ξ^{norm} relation to a Ξ^{lin} relation, where Ξ^{norm} is almost identical to Ξ^{lin} except that the norm of the witness is explicitly given as part of a statement:

$$\Xi^{\mathsf{norm}}_{\hat{n},n,\mu,r,\beta} \coloneqq \left\{ \begin{array}{l} ((\mathbf{H},\mathbf{F},\mathbf{Y},\mathbf{c},\nu),\mathbf{W}) : \\ \quad ((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}) \in \Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta} \\ \quad \|\mathbf{W}\|_{\sigma,2} \leq \nu \leq \beta \end{array} \right\}.$$

The reduction is split into two steps for simplicity: (i) In Section 3.2 we present a RoK $\Pi^{\overline{\mathsf{sum}}}$ which reduces a $\Xi^{\overline{\mathsf{sum}}}$ instance to a $\Xi^{\text{lde-}\otimes}$ instance. $\Xi^{\overline{\mathsf{sum}}}$ is a sumcheck relation defined over $f \cdot \bar{f}$, where f is a low-degree extension of $\mathbf{W}$. (ii) In Section 4.1 we present a RoK Π^{norm} which reduces a Ξ^{norm} instance to a $\Xi^{\overline{\mathsf{sum}}}$ instance. The final results are summarised in Corollary 1.

4.1 Π^{norm}: RoK from Ξ^{norm} to $\Xi^{\text{lde-}\otimes}$

Π^{norm} is a RoK from Ξ^{norm} to $\Xi^{\text{lde-}\otimes}$. The protocol is shown in Fig. 3 and a formal lemma is given in Lemma 4.

Lemma 4. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, \mu \in \mathbb{N}$ and $0 \leq \beta \leq \beta' \leq q$ and $\beta'^2 < q/2$. Π^{norm} is a perfectly correct and knowledge sound reduction of knowledge for*

$$\Xi^{\mathsf{norm}}_{\hat{n},n,\mu,r,\beta} \leftrightarrow \Xi^{\overline{\mathsf{sum}}}_{\hat{n},n,\mu,r,\beta}.$$

The communication cost is $r \log |\mathcal{R}_q|$. Provers run in $\mathcal{O}(r \cdot m)$ ring operations.

Proof. Correctness: Consider a relation $((\mathbf{H}, \mathbf{F}, \mathbf{Y}, \nu), \mathbf{W}) \in \Pi^{\mathsf{norm}}$. Let $\mathbf{W} \coloneqq (\mathbf{w}_i)_{i \in [r]}$. We begin by observing that $\mathsf{Trace}(\langle \mathbf{w}_i, \overline{\mathbf{w}_i} \rangle) = \mathsf{Trace}\left(\sum_{j \in [m]} w_{j,i} \cdot \overline{w_{j,i}}\right) = \sum_{j \in [m]} \sum_{\sigma \in \mathsf{Gal}(\mathcal{R}/\mathbb{Z})} \sigma(w_{j,i} \cdot \overline{w_{j,i}}) = \mathsf{Trace}(t_i) = \nu_i^2$, where $\nu_i = \|\mathbf{w}_i\|_{\sigma,2} \leq \beta$. Since $\mathbf{t} \coloneqq (\langle \mathbf{w}_i, \overline{\mathbf{w}_i} \rangle)$, we have that $((\mathbf{H}, \mathbf{F}, \mathbf{Y}, \mathbf{t}), \mathbf{W}) \in \Xi^{\overline{\mathsf{sum}}}_{\hat{n},n,\mu,r,\beta}$.

Knowledge Soundness: By the definition of $\Xi^{\overline{\mathsf{sum}}}_{\hat{n},n,\mu,r,\beta}$, we have $\sum_{\mathbf{z} \in [d]^\mu} (\mathsf{LDE}[\mathbf{W}] \odot \mathsf{LDE}[\overline{\mathbf{W}}])(\mathbf{z}) = \mathbf{t} \bmod q$. Equivalently, $t'_i = \sum_{j \in [m]} w_{j,i} \cdot \overline{w_{j,i}}$ and $\mathsf{Trace}(t'_i) = \mathsf{Trace}\left(\sum_{j \in [m]} w_{j,i} \cdot \overline{w_{j,i}}\right) \bmod q$, which also holds without mod q since $q/2 > \beta^2$. We conclude that

$$\nu^2 \geq \mathsf{Trace}\left(\sum_{j \in [m]} w_{j,i} \cdot \overline{w_{j,i}}\right) = \sum_{j \in [m]} \sum_{\sigma \in \mathsf{Gal}(\mathcal{R}/\mathbb{Z})} \sigma(w_{j,i} \cdot \overline{w_{j,i}}) = \|\mathbf{w}_i\|^2_{\sigma,2},$$

which implies $(\mathbf{H}, \mathbf{F}, \mathbf{Y}, \nu) \in \Xi^{\mathsf{norm}}$. □

The following corollary is immediate from Lemmas 2 to 4.

Corollary 1. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, \mu \in \mathbb{N}$, $0 \leq \beta \leq \beta' \leq q$ and $\beta'^2 < q/2$. Let $\Pi^{\mathsf{norm+}}$ be a composition of Π^{norm}, $\Pi^{\overline{\mathsf{sum}}}$ and $\Pi^{\mathsf{lde}\text{-}\otimes}$. $\Pi^{\mathsf{norm+}}$ is a perfectly correct reduction of knowledge for: $\Xi^{\mathsf{norm}}_{\hat{n},n,\mu,r,\beta} \rightarrow \Xi^{\mathsf{lin}}_{\hat{n}+2,n+2,r,\mu,\beta}$. It is knowledge sound with knowledge-error $\kappa = {}^{2\mu(d-1)+r-1}/_{q^e}$ for $\Xi^{\mathsf{norm}}_{\hat{n},n,\mu,r,\beta} \cup \Xi^{\mathsf{sis}}_{2\beta'} \leftarrow \Xi^{\mathsf{lin}}_{\hat{n}+2,n+2,r,\mu,\beta'}$. The communication cost is $(2d-1)\mu e \log q + 3r \log |\mathcal{R}_q|$. Prover runs in $\mathcal{O}(rm)$ ring operations.*

5 Construction of SNARK

This section describes how to compose the atomic protocols introduced in [KLNO25] combined with our new protocol Π^{norm} to obtain succinct arguments for the principal relation Ξ^{lin}. The composition is designed to achieve both asymptotic and concrete efficiency, without introducing any soundness gaps.

In Section 5.1 we begin by reviewing the individual functionality of [KLNO25] protocols, which serve as the building blocks for our composition. We then outline the overall composition strategy, including the order of operations required to achieve the desired asymptotic complexity. Furthermore, we provide an analysis of the composition's communication complexity and prover runtime from [KLNO25].

In Section 5.2 we present our enhanced composition strategy, which replaces the $\Pi^{\mathsf{norm\text{-}klno24}}$ protocol with our new Π^{norm} protocol (to be precise, we employ $\Pi^{\mathsf{norm+}}$ from Corollary 1, which we denote here as Π^{norm} for simplicity). We discuss the differences between the original and enhanced protocols, including their respective communication complexities and prover runtime, demonstrating that our approach achieves near-optimal performance. Finally, we provide concrete performance metrics for our composition strategy, including the communication complexity and prover runtime. We present our experimental results, demonstrating the performance.

5.1 Summary of "RoK, paper, SISsors" and "RoK and Roll" [KLNO24, KLNO25]

We recall the core reductions of knowledge from [KLNO24, KLNO25]. These protocols, namely Π^{split}, Π^{fold}, $\Pi^{b\text{-}\mathsf{decomp}}$, Π^{batch}, $\Pi^{\mathsf{norm\text{-}klno24}}$, Π^{join}, and $\Pi^{\otimes\mathsf{RP}}$, serve as the atomic building blocks for constructing succinct arguments over the structured relation Ξ^{lin}. As an important difference, we adapt the norm from the canonical Frobenius norm of $\mathbf{W}$ to the maximum canonical 2-norm over the columns of $\mathbf{W}$. Table 3 gives a visual summary of the properties of these protocols.

Composition. We recall the composition presented in [KLNO25]. The composition strategy is instantiated as follows to achieve asymptotically efficient succinct arguments for the relation $\Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta}$. The looping phase follows the fixed sequence

$$\Pi^{\text{norm-klno24}} \to \Pi^{b\text{-decomp}} \to \Pi^{\mathsf{split}} \to \Pi^{\otimes\mathsf{RP}} \to \begin{matrix}\Pi^{\mathsf{fold}} \\ \Pi^{\mathsf{id}}\end{matrix} \to \Pi^{\mathsf{join}} \to \Pi^{\mathsf{batch}},$$

where Π^{id} denotes the identity protocol, which simply forwards messages without any modification. This sequence is repeated $\mu = \mathcal{O}(\log_\lambda m)$ times. After performing μ iterations of the loop, the witness height is reduced to $\mathcal{O}(\lambda)$. At this point, the protocol turns into the "unstructured loop", where the verifier runtime is no longer asymptotically succinct, which is acceptable since the witness height is now small. The unstructured loop consists of the sequence:

$$\Pi^{\text{norm-klno24}} \to \Pi^{b\text{-decomp}} \to \Pi^{\mathsf{split}} \to \Pi^{\mathsf{RP}} \to \Pi^{\mathsf{fold}} \to\!\to \Pi^{\mathsf{batch}},$$

which is repeated $\mathcal{O}(\log \lambda)$ times until the witness height is reduced to a constant. Finally, the prover sends the remaining witness in the clear.

The communication cost of each loop is dominated by the $\Pi^{\text{norm-klno24}}$, $\Pi^{b\text{-decomp}}$, and Π^{split} protocols, each contributing $\mathcal{O}(1)$ elements from $\mathcal{R}_q$. Summing over all μ rounds yields a total of $\mathcal{O}(\log m)$ $\mathcal{R}_q$ elements. Therefore, the total communication complexity, including the final witness, is $\mathcal{O}\left(\lambda \log^3 m / \log \lambda\right)$ bits.

Prover runtime of [KLNO25] Explicit analysis of the prover's runtime is not included in [KLNO25]. Below, we give a brief overview. Throughout the execution of the protocol, the number of columns in the witness remains bounded by $\mathcal{O}(1)$. The procedures Π^{batch}, Π^{split}, and Π^{fold} consist mainly of linear transformations applied to the witness matrix $\mathbf{W}$ or an auxiliary matrix $\mathbf{F} \in \mathcal{R}_q^{O(1)\times m}$. As a result, each of these steps runs in $\mathcal{O}(m)$ ring operations, which we denote as $\mathcal{O}_\lambda(m)$ for brevity. The same bound also applies to $\Pi^{b\text{-decomp}}$ by a similar argument. The main source of computational overhead is Π^{norm}. This protocol, adapted from the RoK construction in [KLNO24], involves computing a product of two polynomials of degree m. This computation can be performed in $\mathcal{O}_\lambda(m \log m)$ ring operations. Each round of the loop defined above incurs this cost. Since the witness height m shrinks by a constant factor between rounds, the total cost over all rounds forms a geometric series and remains within the same asymptotic bound.

5.2 Replacing the $\Pi^{\text{norm-klno24}}$ with Π^{norm}

We propose a modified "looping phase": $\Pi^{\mathsf{norm}} \to \Pi^{\mathsf{batch}} \to \Pi^{b\text{-decomp}} \to \Pi^{\mathsf{split}} \to \Pi^{\mathsf{fold}}$. Compared to $\Pi^{\text{norm-klno24}}$, the enhanced protocol Π^{norm} offers almost strictly better parameters for the recursive composition. It preserves the norm of the running witness without inflation, avoids increasing the witness width r, and only increases $\underline{n}$ by 1, whereas $\Pi^{\text{norm-klno24}}$ increases it by 3. This makes Π^{norm} almost a drop-in replacement in the composition strategy without affecting the asymptotic communication or runtime analysis. The only deviation lies in the communication cost. In [KLNO25] the cost has been bounded by $\mathcal{O}(1)$ $\mathcal{R}_q$ elements. In our protocol, the cost is bounded by $\mathcal{O}\left(\log |\mathcal{R}_q| + \log m(\lambda/\log \lambda) \log q\right)$. We note that, according to the analysis from [BDGL16], recalled in [KLNO24, KLNO25], the ring degree is assumed $\varphi = \Theta(\lambda \log m / \log \lambda)$ and $\log q = \mathcal{O}(\log m)$. Upon this substitution, both terms become asymptotically equivalent and are simplified to $\mathcal{O}(1)$ $\mathcal{R}_q$ elements.

Total Communication Complexity. After $\mu = \mathcal{O}(\log m)$ rounds, the height of the witness is $\mathcal{O}(1)$ consisting of $\mathcal{R}_q$ elements. The size (in bits) of such a witness is $\mathcal{O}\left(\lambda \cdot \log^2 m / \log \lambda\right)$ The total communication cost across all rounds and the final witness sent is $\mathcal{O}\left(\lambda \log^3 m / \log \lambda\right)$, matching the proof size of [KLNO25].

Prover runtime of enhanced protocol. We observe that throughout the execution of the protocol, the previous analysis of Π^{split}, Π^{fold}, $\Pi^{b\text{-decomp}}$ remains unchanged. For our enhanced Π^{norm}, we observe that due to Lemma 7, prover runtime is also upper-bounded by $\mathcal{O}(m)$ ring operations. We conclude that for each round as defined in the new loop, the prover's runtime is $\mathcal{O}(m)$. The same bound applies to the complete protocol for an arbitrary number of rounds by applying the geometric series as the witness height m shrinks by the constant factor between rounds.

Verifier's Runtime Although not the main focus of this work, we briefly discuss the verifier's runtime, since that analysis is not explicitly provided in [KLNO25]. The main source of complexity in the verifier's runtime comes the execution of $\Pi^{\otimes\mathsf{RP}}$ (and its unstructured counterpart Π^{RP}). The verifier's task in $\Pi^{\otimes\mathsf{RP}}$ involves computing a $\mathcal{O}(\lambda^2)$ operations in $\mathcal{R}_q$, which is the dominant factor in the verifier's runtime. The remaining parts of the protocol contribute a lower-order term to the overall complexity. In total, the verifier's runtime is bounded by $\mathcal{O}(\log m\lambda^2)$ ring operations, due to the $\mathcal{O}(\log m)$ rounds of the loop.

5.3 Putting Everything Together

The deductions above lead to the following main theorem, which shall be viewed as a main result of this paper. We state the theorem informally, omitting the knowledge error and correctness error for simplicity, since the parameters are set to have both errors negligible. We rely on the vSIS assumption, which already well-established in the literature [CLM23] and for completeness, we recall its definition in Appendix A, precisely in Definition 5. The problem is defined for a relation $\Xi^{\mathsf{lin}}_{\mathsf{params},\beta}$ consisting of instances $(\mathbf{F}, \mathbf{0})$ and witnesses $\mathbf{w}$ such that $\mathbf{F} \in \mathcal{R}_q^{n\times m}$ is randomly sampled. Witness satisfies the linear equation $\mathbf{F}\cdot\mathbf{w} = \mathbf{0} \pmod q$ and the norm bound $\|\mathbf{w}\|_2 \leq \beta$. The problem is assumed to be as hard as regular ring variant of SIS, i.e. for $\mathbf{F}$ sampled uniformly from $\mathcal{R}_q^{n\times m}$ for $m = d^\mu$.

Theorem 1. *(Informal) Let* $\mathsf{params} = (\mathcal{R}, q, n, \mu)$*, where* $\mathcal{R}$ *be a ring,* $q \in \mathbb{N}$ *a modulus,* $n \in \mathbb{N}$ *dimensions,* $0 < \beta \ll \beta_{\mathsf{vSIS}} < q/2$ *a norm bounds,* $\mu \in \mathbb{N}$ *parameter. Assuming the* $\mathsf{vSIS}_{\mathsf{params},\beta_{\mathsf{vSIS}}}$ *assumption, there exists a succinct argument of knowledge for the relation* $\Xi^{\mathsf{lin}}_{\mathsf{params},\beta}$ *with negligible correctness and knowledge error (in the security parameter), or more precisely, reduction of knowledge for,*

$$\Xi^{\mathsf{lin}}_{\mathsf{params},\beta} \hookleftarrow \{0/1\},$$

where $\{0/1\}$ *is a trivial relation that accepts or rejects unconditionally. The reduction is characterised by the following complexities:*

- *Proof Size:* $\mathcal{O}\left(\lambda \log^3 m/\log\lambda\right)$ *bits*
- *Prover time:* $\mathcal{O}(m)$ *ring operations*
- *Verifier time:* $\mathcal{O}(\log m\lambda^2)$ *ring operations*

5.4 Instantiation of Polynomial Commitment Scheme (PCS)

Without formal definition, we briefly discuss how to instantiate a multilinear polynomial commitment scheme (PCS) over $\mathcal{R}_q$ from an argument system for the relation Ξ^{lin}. We focus only on the multilinear case as it is strictly more general than the univariate case.

A multilinear polynomial commitment scheme (PCS) allows a prover to commit to a multilinear polynomial and later open the commitment by providing the evaluation of the polynomial at a specific point. The argument system for the relation Ξ^{lin} can be used to construct the PCS by leveraging the properties of the argument system to ensure soundness and completeness of the commitment scheme.

The construction is a direct consequence of the $\Pi^{\mathsf{lde}\text{-}\otimes}$ protocol presented in Lemma 2. In words, upon commitment to the coefficients of the polynomial ($\mathbf{w} \in \mathcal{R}_q^{d^\mu}$), the prover open the commitment to the evaluation point $\mathbf{x}^\intercal := (x_i)_{i\in[\mu]}$ so that $\mathsf{LDE}[\mathbf{w}](\mathbf{x}) = t$. Such a relation is expressed as an instance of $\Xi^{\mathsf{lde}\text{-}\otimes}$ and reduced to Ξ^{lin} by Lemma 2. The prover and verifier can then run the argument system for Ξ^{lin} to prove the correctness of the opening.

The aforementioned construction inherits the properties of the argument system for Ξ^{lin}, including knowledge soundness and efficiency. The communication complexity, prover time, and verifier time of the PCS will be directly related to those of the argument system.

We summarise the above discussion in the following theorem.

Theorem 2. *(Informal) Let* $\mathsf{params} = (\mathcal{R}, q, n, \mu)$*, where* $\mathcal{R}$ *be a ring,* $q \in \mathbb{N}$ *a modulus,* $n \in \mathbb{N}$ *dimensions,* $0 < \beta \ll \beta_{\mathsf{vSIS}} < q/2$ *a norm bounds,* $\mu \in \mathbb{N}$ *parameter. Assuming the* $\mathsf{vSIS}_{\mathsf{params},\beta_{\mathsf{vSIS}}}$ *assumption, there exists a norm-bounded (in the sense of the canonical 2-norm of the polynomial coefficient bounded by* β*) multilinear polynomial commitment scheme (PCS) over* $\mathcal{R}_q$ *that is correct and knowledge-sound with negligible correctness and knowledge error (in the security parameter), The PCS is characterised by the following complexities:*

– *Commitment size:* $\mathcal{O}\left(\lambda \log^2 m / \log \lambda\right)$ *bits*
– *Commitment time:* $\mathcal{O}(m)$ *ring operations*
– *Opening size:* $\mathcal{O}\left(\lambda \log^3 m / \log \lambda\right)$ *bits*
– *Prover time:* $\mathcal{O}(m)$ *ring operations*
– *Verifier time:* $\mathcal{O}(\log m \lambda^2)$ *ring operations*

5.5 Parameters Selection and Experimental Results

We provide a concrete instantiation of the parameters for the argument system for Ξ^{lin} and the resulting PCS (being exactly the same constraints system). Table 1 (moved to the Section 1 for visibility) summarises the results of our experiments. We conducted benchmarks parametrised by different number of $\mathbb{Z}_q$ elements in the witness, i.e. $m \cdot \varphi \in \{2^{26}, 2^{28}, 2^{30}\}$ and presented them together with results imported from other popular works. We present the protocol in 3 variant differing in the degree of the ring $\mathcal{R}_q$, i.e. $\varphi \in \{2^7, 2^8, 2^9\}$, which serves as trade-off between the proof size and the prover/verifier time. The number of rounds has been selected from $\mu \in [9, 15]$ (depending of the length of the witness m) with the last 3 rounds being unstructured. Further details about the implementation and parameters selection (mainly shared with the folding scheme instantiation) are provided in Appendix E.

6 Construction of Folding Scheme

In this section, we show the construction of a folding scheme for the linear relation Ξ^{lin} composed of six RoKs, namely, $\Pi^{\mathsf{norm}}, \Pi^{\otimes\mathsf{RP}}, \Pi^{\mathsf{fold}}, \Pi^{\mathsf{join}}, \Pi^{\mathsf{batch}}$, and $\Pi^{b\text{-}\mathsf{decomp}}$.

A folding scheme is a reduction of knowledge (RoK) that takes as input witness-statement pairs of a relation and outputs a single instance of the same relation. To formalise the intuition, we define the following relation, which is a slight modification of Ξ^{lin} where the commitment matrix $\overline{\mathbf{F}}$ is viewed as a part of the public parameter.

$$\Xi^{\mathsf{lin\text{-}pub}}_{\hat{n},n,\mu,r,\beta,\overline{\mathbf{F}}} := \left\{ \begin{array}{l} ((\mathbf{H}, \underline{\mathbf{F}}, \mathbf{Y}), \mathbf{W}) \\ \quad \mathbf{H} \in \mathcal{R}_q^{\hat{n} \times n} \quad \mathbf{F} = \begin{pmatrix} \overline{\mathbf{F}} \\ \underline{\mathbf{F}} \end{pmatrix} \in \mathcal{R}_q^{n \times m} \\ \quad \mathbf{Y} \in \mathcal{R}_q^{\hat{n} \times r} \quad \mathbf{W} \in \mathcal{R}^{m \times r} \\ \quad \|\mathbf{W}\|_{\sigma,2} \leq \beta \;\text{ and }\; \mathbf{HFW} = \mathbf{Y} \bmod q \end{array} \right\}$$

We define analogously $\Xi^{\mathsf{lin\text{-}pub\text{-}}\otimes}$ where the matrix $\mathbf{F}$ is required to be a row-tensor matrix.

This special relation allows us to formalise the folding scheme theorem.

Theorem 3. *(Informal) Let* $\mathsf{params} = (\mathcal{R}, q, n, \mu, \overline{\mathbf{F}})$*, where* $\mathcal{R}$ *be a ring,* $q \in \mathbb{N}$ *a modulus,* $n, \overline{n} \in \mathbb{N}$ *dimensions,* $0 < \beta \ll \beta_{\mathsf{vSIS}} < q/2$ *a norm bounds,* $\mu \in \mathbb{N}$ *parameter, and* $\mathbf{F} \in \mathcal{R}_q^{\overline{n} \times m}$ *be matrices. Assuming the* $\mathsf{vSIS}_{\mathsf{params},\beta_{\mathsf{vSIS}}}$ *assumption, there exists a folding scheme for the relation* $\Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta}$ *with negligible soundness and correctness errors. More, precisely, there exists a reduction of knowledge* Π^{fs} *for*

$$(\Xi^{\mathsf{lin\text{-}pub\text{-}}\otimes}_{\hat{n},n,\mu,r,\beta,\mathbf{F}})^{r_{\mathsf{acc}}+L} \longleftrightarrow (\Xi^{\mathsf{lin\text{-}pub\text{-}}\otimes}_{\hat{n},n+(n-\overline{n})\cdot(L-1)+4,\mu,r,\beta,\mathbf{F}})^{r_{\mathsf{acc}}},$$

for $r_{\mathsf{acc}} = 2\ell$*, where* ℓ *is the decomposition parameter, and* L *is the number of instances to be folded. Let* $L, n = \mathcal{O}(1)$*. The folding scheme is characterised by the following complexities:*

– *Proof size:* $\mathcal{O}\left(\lambda \log^2 m / \log \lambda\right)$ *bits*
– *Prover time:* $\mathcal{O}(m)$ *ring operations*
– *Verifier time:* $\mathcal{O}(\lambda^2)$ *ring operations*

The following section describes the building blocks of the folding scheme and their composition, serving as a proof of Theorem 3.

6.1 Composition of RoKs

The composition of the RoKs for folding scheme is strikingly similar to the one for a single round of structured round of the argument system in Section 5, i.e. it consists of the following six individual RoKs:

$$\Pi^{\mathsf{join}} \to \Pi^{\mathsf{norm}} \to \Pi^{\otimes\mathsf{RP}} \to \begin{matrix}\Pi^{\mathsf{fold}} \\ \Pi^{\mathsf{id}}\end{matrix} \to \Pi^{\mathsf{join}} \to \Pi^{\mathsf{batch}} \to \Pi^{b\text{-}\mathsf{decomp}},$$

The main difference is that the Π^{split} RoK is missing since the relation does not need to be split into smaller relations for the folding scheme. Furthermore, the chain of the reduction is prefixed with an additional Π^{join} RoK, which merges the multiple instances to be folded into a single relation. This requires sending cross-terms of the bottom rows of $\mathbf{F}$ corresponding to different instances. As a result, the number of rows of $\mathbf{F}$ increases by $(n - \overline{n}) \cdot (L - 1)$ where L is the number of instances to be folded.

We carefully track the parameters of the relation throughout the composition. We start with the relation $\Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r_{\mathsf{acc}}+L,\beta}$ where L is the number of instances to be folded. The first RoK Π^{norm} takes as input the relation $\Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r_{\mathsf{acc}}+L,\beta}$ and outputs the relation $\Xi^{\mathsf{lin}}_{\hat{n}+2,n+2,\mu,r_{\mathsf{acc}}+L,\beta}$. The protocol does not impact the parameters otherwise, but serves as a "checkpoint" to ensure that the witness norm is bounded by β in the extraction direction. Next, the protocol $\Pi^{\otimes\mathsf{RP}}$ takes as input the relation $\Xi^{\mathsf{lin}}_{\hat{n}+2,n+2,\mu,r_{\mathsf{acc}}+L,\beta}$ and outputs two relations: $\Xi^{\mathsf{lin}}_{\hat{n}+3,n+3,\mu,r_{\mathsf{acc}}+L,\beta}$ and $\Xi^{\mathsf{lin}}_{\hat{n}+1,n+1,1,\hat{\beta}}$ where $\hat{\beta} = m_{\mathsf{rp}} \cdot \beta$ and $m_{\mathsf{rp}} = \mathcal{O}(\lambda)$. This projection will be used to argue about the approximate norm of the witness in the extraction direction. The next RoK Π^{fold} takes the first relation $\Xi^{\mathsf{lin}}_{\hat{n}+3,n+3,\mu,r_{\mathsf{acc}}+L,\beta}$ and outputs the relation $\Xi^{\mathsf{lin}}_{\hat{n}+3,n+3,\mu,1,(r_{\mathsf{acc}}+L)\gamma\beta}$ where γ is an expansion factor of the challenge set. Two branches are then merged using the RoK Π^{join}, resulting in the relation $\Xi^{\mathsf{lin}}_{\hat{n}+4,n+4,\mu,2,\max(\hat{\beta},(r_{\mathsf{acc}}+L)\gamma\beta)}$. Eventually, the last two RoKs Π^{batch} and $\Pi^{b\text{-}\mathsf{decomp}}$ take as input the relation $\Xi^{\mathsf{lin}}_{\hat{n}+4,n+4,\mu,2,\max(\hat{\beta},(r_{\mathsf{acc}}+L)\gamma\beta)}$ and output the relation $\Xi^{\mathsf{lin}}_{\hat{n},n+4,\mu,2\ell,\beta}$ as ℓ is set so that decomposition of the witness of norm bounded by $\max(\hat{\beta}, (r_{\mathsf{acc}} + L)\gamma\beta)$ results in a witness of norm bounded by β.

The asymptotic analysis follows directly from the asymptotic analysis of the individual RoKs, assuming that $L, \ell, \underline{n} = \mathcal{O}(1)$.

6.2 Parameters Selection and Experimental Results

We provide a concrete instantiation of the parameters for the folding scheme for Ξ^{lin}. We set the decomposition parameter $\ell = 2$ so that the accumulator consists of four columns. For simplicity, consider the case where the number of instances to be folded is $L = 4$ (so that the number of columns in total for the input relation is 8). We consider uniform folding of exactly the same ISIS-like statements (with different images), which are expressed as the relation $((\mathbf{F}, \mathbf{W}), \mathbf{Y}) \in \Xi^{\mathsf{lin}}_{\hat{n},n,\mu,8,\beta}$, where n is set so that the ISIS instance is hard

We set the degree of the cyclotomic ring to 128, i.e. we consider 256-th cyclotomic ring. We consider 3 cases with the number rows the witness set as $m \in \{2^{17}, 2^{19}, 2^{21}\}$, which correspond to folding $\{2^{19}, 2^{21}, 2^{23}\}$ ring elements (distributed over 4 columns) with fixed accumulator.

The results of the experiments are presented in Table 2. The results show that the folding scheme is efficient in practice, with the proof size being in the order of less than 75 KBs and the prover time being in the order of a few seconds even for large instances. The verifier time is also very efficient, being in the order of milliseconds. Further details about the implementation and parameters selection (mainly shared with the argument system instantiation) are provided in Appendix E.

m	$\mathbb{Z}_q$ elements	Proof	$\mathcal{P}$	$\mathcal{V}$
2^{17}	2^{26}	70.1 KB	0.45 s	2.18 ms
2^{19}	2^{28}	72.4 KB	1.66 s	2.28 ms
2^{21}	2^{30}	72.5 KB	5.52 s	2.51 ms

Table 2: Experimental results for the folding scheme for the relation Ξ^{lin} with $L = 4$ instances to be folded, $\ell = 2$, and ring degree 128.

Our results are difficult to compare with, e.g. state of the art folding schemes for lattice-based relations, such as [BC25b], since they consider slightly different and more general relations. Further, the results from [BC25b] are not implemented and our preliminary experiments show that the performance is not competitive with our folding scheme due to the expensive computation of the "double commitments"[9].

6.3 Improving the Folding Scheme via Enhanced Batching

We remark that the folding scheme can be improved by replacing the Π^{batch} RoK with a more efficient batching protocol. In particular, the aforementioned analysis relies on the assumption that the number of "unbatched" rows of $\mathbf{F}$ is constant, i.e. $n, L = \mathcal{O}(1)$. This is because although the Π^{batch} RoK reduces the number of output rows of $\mathbf{H}$ from $\hat{n}+4$ to $\hat{n}$, the verifier and prover runtimes of the protocol scale linearly in n. As a consequence, the analysis shall be restricted to the case where the number of folding rounds is constant (since in each round, the number of rows of $\mathbf{F}$ increases by a constant).

To address this issue, we propose an alternative batching protocol based on the sumcheck protocol. The protocol instead of naively batching the rows of $\mathbf{F}$, presents the relation $\underline{\mathbf{F}}\mathbf{W} = \underline{\mathbf{Y}} \bmod q$ as a set of sumcheck claims, which are batched and reduced to a single evaluation claim. This trick effectively restores the structure of the relation, so that the performance of the folding scheme does not degrade as the number of folding rounds increases. The details of the protocol Π^{batch^*} appear in Appendix D.

The aforementioned improvement yields the following theorem as a corollary of Theorem 3.

Theorem 4. *(Informal) For exactly the same setting as in Theorem 3, there exists a reduction of knowledge Π^{fs^+} for*

$$(\Xi^{\mathsf{lin\text{-}pub\text{-}}\otimes}_{\hat{n},n,\mu,r,\beta,\mathbf{F}})^{r_{\mathsf{acc}}+L} \leftrightarrow (\Xi^{\mathsf{lin\text{-}pub\text{-}}\otimes}_{\hat{n},n,\mu,r,\beta,\mathbf{F}})^{r_{\mathsf{acc}}},$$

which is obtained by replacing the Π^{batch} RoK in Π^{fs} with the Π^{batch^*} RoK. The folding scheme is characterised by the same complexities as in Theorem 3.

Remark 2. We remark that the rows of $\mathbf{F}$, except those newly added throughout the execution of the protocol, are not required to be row-tensors as they are not processed by the Π^{fs^+} or Π^{fs} RoKs. If we ignore the structure, then the folding scheme could be applied to a more general relation and without dependency on the vSIS assumption. However, after folding the relation, the resulting relation would not be compatible with the argument system in Section 5.

7 Supporting Rank-1 Constraint Systems

In this section we remark the committed rank-1 constraint system (R1CS) relations $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ and the RoK from $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ to the principal relation Ξ^{lin}. The `R1CS` relation is widely-used to capture an arithmetic computation. The committed `R1CS` relation is defined as follows:

Definition 4 ($\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$: Committed `R1CS` Relations). *The committed* `R1CS` *relation $\Xi^{\mathsf{lin\text{-}r1cs}}$ is defined as*

$$\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta} := \left\{ \begin{array}{l} ((\mathbf{H},\mathbf{F},\mathbf{Y},\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{E}),\mathbf{W}) \\ \quad ((\mathbf{H},\mathbf{F},\mathbf{Y}),\mathbf{W}) \in \Xi^{\mathsf{lin}}_{\hat{n},n,\mu,r,\beta} \\ \quad \mathbf{A},\mathbf{B},\mathbf{C} \in \mathcal{R}_q^{\tilde{m}\times m},\ \mathbf{D} \in \mathcal{R}_q^{\tilde{n}\times d^{\otimes\mu}},\ \mathbf{E} \in \mathcal{R}_q^{\tilde{n}\times r} \\ \quad \mathbf{AW} \odot \mathbf{BW} = \mathbf{CW} \bmod q,\ \mathbf{DW} = \mathbf{E} \bmod q \end{array} \right\}.$$

The structured variant $\Xi^{\mathsf{lin\text{-}r1cs}\text{-}\otimes}$ is identical to $\Xi^{\mathsf{lin\text{-}r1cs}}$, except that the matrices $\mathbf{A},\mathbf{B},\mathbf{C}$ in the statement are required to be in $\mathcal{R}_q^{d^{\otimes\tilde{\mu}}\times d^{\otimes\mu}}$.

We note that the requirement of $\mathbf{D} \in \mathcal{R}_q^{\tilde{n}\times d^{\otimes\mu}}$ is not overly restrictive. Indeed, a typical formulation of `R1CS` requires that the witness $\mathbf{w}$ is of the form $\mathbf{w} = (1, \bar{\mathbf{w}})$ and satisfies $\mathbf{Aw} \odot \mathbf{Bw} = \mathbf{Cw}$ (without any extra linear constraint). This requirement of the witness can be checked by ensuring $(1,0,\ldots,0)\mathbf{w} = 1$,

[9] Computation of "double commitment" requires factor of $\varphi \log_\varphi B$ more ring operations compared to a standard Ajtai commitment, where B is the bound on the ℓ_∞-norm of the witness. On the other hand, those operations are mainly additions, which are individually cheaper in practice, but still the overhead is significant.

where $(1, 0, \ldots, 0)$ can be written as a tensor product if the dimension is appropriate. Moreover, a witness $\mathbf{W}$ in a committed R1CS relation $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ is required to have a norm bounded by β.

The following lemma captures the folklore reduction from R1CS to linear relations based on linearisation [KS24, BC25b] (generalised to low-degree setting). The reduction is illustrated in Fig. 6.

Lemma 5. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, \mu, d, \tilde{d} \in \mathbb{N}$ and $0 \leq \beta \leq \beta' \leq q$. $\Pi^{\mathsf{lin\text{-}r1cs}}$ is a perfectly correct reduction of knowledge for:*

$$\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta} \to \Xi^{\mathsf{lde}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta,3}.$$

It is knowledge sound with knowledge-error $(\tilde{m}-1)/q^e + \frac{r\varphi/e-1+\tilde{\mu}3(d-1)}{q^e}$ for

$$\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta'} \cup \Xi^{\mathsf{sis}}_{2\beta'} \leftarrow \Xi^{\mathsf{lde}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta',3}.$$

The communication cost is $(3(d-1)+1)\tilde{\mu}e \log q + 3r \log |\mathcal{R}_q|$. Prover runs in $\mathcal{O}(m\tilde{m}r)$ ring operations.

Appendix C serves as a proof sketch of Lemma 5.

References

AAB+24. Marius A. Aardal, Diego F. Aranha, Katharina Boudgoust, Sebastian Kolby, and Akira Takahashi. Aggregating falcon signatures with LaBRADOR. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 71–106. Springer, Cham, August 2024. 1

ABPS24. Shahla Atapoor, Karim Baghery, Hilder V. L. Pereira, and Jannik Spiessens. Verifiable FHE via lattice-based SNARKs. *CiC*, 1(1):24, 2024. 1

ACK21. Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 549–579, Virtual Event, August 2021. Springer, Cham. 1, 26

ACL+22. Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 102–132. Springer, Cham, August 2022. 5

AFLN24. Martin R. Albrecht, Giacomo Fenzi, Oleksandra Lapiha, and Ngoc Khanh Nguyen. SLAP: Succinct lattice-based polynomial commitments from standard assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VII*, volume 14657 of *LNCS*, pages 90–119. Springer, Cham, May 2024. 1, 5

AHIV17. Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, October / November 2017. 4

AHIV23. Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: lightweight sublinear arguments without a trusted setup. *DCC*, 91(11):3379–3424, 2023. 1, 5

AL21. Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 519–548, Virtual Event, August 2021. Springer, Cham. 1

ALS20. Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 470–499. Springer, Cham, August 2020. 5, 36, 37

APS15. Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. 36

BBB+18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018. 1

BBHR18. Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl, July 2018. 1, 4, 5

BC25a. Dan Boneh and Binyi Chen. Latticefold: A lattice-based folding scheme and its applications to succinct proof systems. 2025. 1, 2, 7

BC25b. Dan Boneh and Binyi Chen. LatticeFold+: Faster, simpler, shorter lattice-based folding for succinct proof systems. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 327–361. Springer, Cham, August 2025. 1, 2, 3, 19, 20, 36

BCFW25. Benedikt Bünz, Alessandro Chiesa, Giacomo Fenzi, and William Wang. Linear-time accumulation schemes. Cryptology ePrint Archive, Paper 2025/753, 2025. 1

BCG+19. Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, and Nicholas Spooner. Linear-size constant-query IOPs for delegating computation. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 494–521. Springer, Cham, December 2019. 5

BCHO22. Jonathan Bootle, Alessandro Chiesa, Yuncong Hu, and Michele Orrù. Gemini: Elastic SNARKs for diverse environments. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 427–457. Springer, Cham, May / June 2022. 5

BCPS18. Anurag Bishnoi, Pete L. Clark, Aditya Potukuchi, and John R. Schmitt. On zeros of a polynomial in a finite grid. *Combinatorics, Probability and Computing*, 27(3):310–333, 2018. 25

BCR+19. Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Cham, May 2019. 1, 5

BCS16. Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Berlin, Heidelberg, October / November 2016. 5

BCS21. Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Sumcheck arguments and their applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 742–773, Virtual Event, August 2021. Springer, Cham. 1

BCS23. Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Lattice-based succinct arguments for NP with polylogarithmic-time verification. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 227–251. Springer, Cham, August 2023. 5

BCTV14. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 276–294. Springer, Berlin, Heidelberg, August 2014. 2

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th SODA*, pages 10–24. ACM-SIAM, January 2016. 15

BFS20. Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent SNARKs from DARK compilers. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 677–706. Springer, Cham, May 2020. 5

BGH19. Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Report 2019/1021, 2019. 2

BK20. Dan Boneh and Sam Kim. One-time and interactive aggregate signatures from lattices. https://crypto.stanford.edu/~skim13/agg_ots.pdf, 2020. 1

BL25. Katharina Boudgoust and Oleksandra Lapiha. Leftover hash lemma(s) over cyclotomic rings. ASIACRYPT 2025, to appear, 2025. 36

BLNS20. Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 441–469. Springer, Cham, August 2020. 1

BLS19. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 176–202. Springer, Cham, August 2019. 5

BMNW25a. Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang. Accumulation without homomorphism. In Raghu Meka, editor, *ITCS 2025*, volume 325, pages 23:1–23:25. LIPIcs, January 2025. 1

BMNW25b. Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang. Arc: Accumulation for reed-solomon codes. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 128–160. Springer, Cham, August 2025. 1

BS23. Ward Beullens and Gregor Seiler. LaBRADOR: Compact proofs for R1CS from module-SIS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 518–548. Springer, Cham, August 2023. 1, 2, 5, 36

CBBZ23. Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. HyperPlonk: Plonk with linear-time prover and high-degree custom gates. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 499–530. Springer, Cham, April 2023. 3, 5, 12

CHK+21. Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. NTT multiplication for NTT-unfriendly rings. *IACR TCHES*, 2021(2):159–188, 2021. 4

CHM+20. Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 738–768. Springer, Cham, May 2020. 1, 5

CLM23. Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Lattice-based succinct arguments from vanishing polynomials - (extended abstract). In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 72–105. Springer, Cham, August 2023. 1, 2, 5, 6, 16, 25

CMNW24. Valerio Cini, Giulio Malavolta, Ngoc Khanh Nguyen, and Hoeteck Wee. Polynomial commitments from lattices: Post-quantum security, fast verification and transparent setup. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 207–242. Springer, Cham, August 2024. 1, 4, 5

CT10. Alessandro Chiesa and Eran Tromer. Proof-carrying data and hearsay arguments from signature cards. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 310–331. Tsinghua University Press, January 2010. 2

DKLW25. Adrien Dubois, Michael Klooß, Russell W. F. Lai, and Ivy K. Y. Woo. Lattice-based proof-friendly signatures from vanishing short integer solutions. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part I*, volume 15674 of *LNCS*, pages 452–486. Springer, Cham, May 2025. 1

FKNP24. Giacomo Fenzi, Christian Knabenhans, Ngoc Khanh Nguyen, and Duc Tu Pham. Lova: Lattice-based folding scheme from unstructured lattices. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 303–326. Springer, Singapore, December 2024. 1, 2

FLV23. Ben Fisch, Zeyu Liu, and Psi Vesely. Orbweaver: Succinct linear functional commitments from lattices. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 106–131. Springer, Cham, August 2023. 5

GBK+24. Mariana Gama, Emad Heydari Beni, Jiayi Kang, Jannik Spiessens, and Frederik Vercauteren. Blind zkSNARKs for private proof delegation and verifiable computation over encrypted data. Cryptology ePrint Archive, Report 2024/1684, 2024. 1

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Berlin, Heidelberg, May 2013. 5

GLS+23. Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic SNARKs for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 193–226. Springer, Cham, August 2023. 1, 4, 5

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Berlin, Heidelberg, May 2016. 1, 5

GSW23. Riddhi Ghosal, Amit Sahai, and Brent Waters. Non-interactive publicly-verifiable delegation of committed programs. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 575–605. Springer, Cham, May 2023. 1

GWC19. Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. 5

HSS24. Intak Hwang, Jinyeong Seo, and Yongsoo Song. Concretely efficient lattice-based polynomial commitment from standard assumptions. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 414–448. Springer, Cham, August 2024. 4, 5

HYJ+25. Syed Mahbub Hafiz, Bahattin Yildiz, Marcos A. Simplicio Jr, Thales B. Paiva, Henrique Ogawa, Gabrielle De Micheli, and Eduardo L. Cominetti. Incompleteness in number-theoretic transforms: New tradeoffs and faster lattice-based cryptographic applications. Cryptology ePrint Archive, Paper 2025/768, 2025. 4

JL25. Kalle Jyrkinen and Russell W. F. Lai. Vanishing short integer solution, revisited - reductions, trapdoors, homomorphic signatures for low-degree polynomials. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part II*, volume 15675 of *LNCS*, pages 273–300. Springer, Cham, May 2025. 2

JRS23. Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders. Lattice signature with efficient protocols, application to anonymous credentials. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 351–383. Springer, Cham, August 2023. 1

JRS24. Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders. Phoenix: Hash-and-sign with aborts from lattice gadgets. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part I*, pages 265–299. Springer, Cham, June 2024. 1

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992. 1

KLNO24. Michael Klooß, Russell W. F. Lai, Ngoc Khanh Nguyen, and Michal Osadnik. RoK, paper, SISsors toolkit for lattice-based succinct arguments - (extended abstract). In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part V*, volume 15488 of *LNCS*, pages 203–235. Springer, Singapore, December 2024. 1, 2, 3, 5, 6, 9, 13, 14, 15, 25, 32, 33, 34, 36

KLNO25. Michael Klooß, Russell W. F. Lai, Ngoc Khanh Nguyen, and Michal Osadnik. Rok and roll – verifier-efficient random projection for $\tilde{O}(\lambda)$-size lattice arguments. In *ASIACRYPT 2025*, 2025. 1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 25, 33, 34, 36

KP16. Yael Tauman Kalai and Omer Paneth. Delegating RAM computations. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 91–118. Springer, Berlin, Heidelberg, October / November 2016. 1

KP23. Abhiram Kothapalli and Bryan Parno. Algebraic reductions of knowledge. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 669–701. Springer, Cham, August 2023. 2

KS24. Abhiram Kothapalli and Srinath T. V. Setty. HyperNova: Recursive arguments for customizable constraint systems. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 345–379. Springer, Cham, August 2024. 7, 20

Lee21. Jonathan Lee. Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 1–34. Springer, Cham, November 2021. 1, 5

LFKN92. Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992. 3

Lib24. Benoît Libert. Simulation-extractable KZG polynomial commitments and applications to HyperPlonk. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14602 of *LNCS*, pages 68–98. Springer, Cham, April 2024. 5

LLZ+25. Fengrun Liu, Haofei Liang, Tianyu Zhang, Yuncong Hu, Xiang Xie, Haisheng Tan, and Yu Yu. HasteBoots: Proving FHE bootstrapping in seconds. Cryptology ePrint Archive, Report 2025/261, 2025. 1

LM23. Russell W. F. Lai and Giulio Malavolta. Lattice-based timed cryptography. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 782–804. Springer, Cham, August 2023. 1, 2

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Cham, August 2022. 5

LNSW13. San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer, Berlin, Heidelberg, February / March 2013. 5

LS18. Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Cham, April / May 2018. 36

LSS+21. Zhichuang Liang, Shiyu Shen, Yuantao Shi, Dongni Sun, Chongxuan Zhang, Guoyun Zhang, Yunlei Zhao, and Zhixiang Zhao. Number theoretic transform: Generalization, optimization, concrete analysis and applications. In Yongdong Wu and Moti Yung, editors, *Information Security and Cryptology*, pages 415–432, Cham, 2021. Springer International Publishing. 4

LSS24. Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer. The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 3125–3137. ACM Press, October 2024. 4

MBKM19. Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2111–2128. ACM Press, November 2019. 5

Mic94. Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994. 1

NS24. Ngoc Khanh Nguyen and Gregor Seiler. Greyhound: Fast polynomial commitments from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 243–275. Springer, Cham, August 2024. 1, 4, 5, 36

NS25. Wilson Nguyen and Srinath Setty. Neo: Lattice-based folding scheme for CCS over small fields and pay-per-bit commitments. Cryptology ePrint Archive, Report 2025/294, 2025. 1, 2

OKC+25. Michal Osadnik, Darya Kaviani, Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Papercraft: Lattice-based verifiable delay function implemented. In Marina Blanton, William Enck, and Cristina Nita-Rotaru, editors, *2025 IEEE Symposium on Security and Privacy*, pages 1603–1621. IEEE Computer Society Press, May 2025. 1, 2, 3, 4

PHGR13. Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. 1, 5

PMH+25. Thales B. Paiva, Gabrielle De Micheli, Syed Mahbub Hafiz, Marcos A. Simplicio Jr., and Bahattin Yildiz. Faster amortized bootstrapping using the incomplete NTT for free. Cryptology ePrint Archive, Paper 2025/696, 2025. 4

Set20. Srinath Setty. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 704–737. Springer, Cham, August 2020. 5

Tha13. Justin Thaler. Time-optimal interactive proofs for circuit evaluation. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 71–89. Springer, Berlin, Heidelberg, August 2013. 3, 12

Val08. Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, March 2008. 2

WTs+18. Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. Doubly-efficient zkSNARKs without trusted setup. In *2018 IEEE Symposium on Security and Privacy*, pages 926–943. IEEE Computer Society Press, May 2018. 5

WW23. Hoeteck Wee and David J. Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 201–235. Springer, Singapore, December 2023. 5

XHY24. Jie Xie, Yuncong Hu, and Yu Yu. Hadamard product argument from lagrange-based univariate polynomials. In Tianqing Zhu and Yannan Li, editors, *ACISP 24, Part I*, volume 14895 of *LNCS*, pages 472–492. Springer, Singapore, July 2024. 5

XZZ+19. Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 733–764. Springer, Cham, August 2019. 3, 5, 12

ZLW+25. Zhelei Zhou, Yun Li, Yuchen Wang, Zhaomin Yang, Bingsheng Zhang, Cheng Hong, Tao Wei, and Wenguang Chen. ZHE: Efficient Zero-Knowledge Proofs for HE Evaluations . In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 3087–3105, Los Alamitos, CA, USA, May 2025. IEEE Computer Society. 1

Protocol	Correctness	Extraction	Communication
Π^{split}	$\Xi^{\mathsf{lin}}_{m,r,\beta} \mapsto \Xi^{\mathsf{lin}}_{m/d,r\cdot d,\beta}$	$\Xi^{\mathsf{lin}}_{m,\sqrt{d}\beta'} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{m/d,r\cdot d,\beta'}$	$(d-1)r\hat{n}\log\lvert\mathcal{R}_q\rvert$
Π^{fold}	$\Xi^{\mathsf{lin}}_{m,r,\beta} \mapsto \Xi^{\mathsf{lin}}_{m,r\gamma\beta}$	$\Xi^{\mathsf{lin}}_{m,r,\beta'} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{m,\beta'}$	0
$\Pi^{b\text{-}\mathsf{decomp}}$	$\Xi^{\mathsf{lin}}_{r,\beta} \mapsto \Xi^{\mathsf{lin}}_{r\ell,\frac{1}{2}\sqrt{m}\sqrt{\mathsf{f}\varphi}b}$	$\Xi^{\mathsf{lin}}_{r,\frac{b^\ell-1}{b-1}\beta'} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{r\ell,\beta'}$	$(\ell-1)r\hat{n}\log\lvert\mathcal{R}_q\rvert$
Π^{batch}	$\Xi^{\mathsf{lin}}_{\hat{n},\beta} \mapsto \Xi^{\mathsf{lin}}_{\overline{n}+1,\beta}$	$\Xi^{\mathsf{lin}}_{\hat{n},\beta'} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{\overline{n}+1,\beta'}$	0
$\Pi^{\mathsf{norm\text{-}klno24}}$	$\Xi^{\mathsf{lin}}_{m,\hat{n},r,\beta} \mapsto \Xi^{\mathsf{lin}}_{m,\hat{n}+3,r+\ell,\tilde{\beta}}$	$\Xi^{\mathsf{lin}}_{m,\hat{n},r,\sqrt{r}\beta} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{m,\hat{n}+3,r+\ell,\tilde{\beta}'}$	$(3\cdot(\ell+r)+\ell\hat{n})\log\lvert\mathcal{R}_q\rvert$
Π^{join}	$\Xi^{\mathsf{lin}}_{\hat{n}_0,n_0,r_0}\times\Xi^{\mathsf{lin}}_{\hat{n}_1,n_1,r_1} \mapsto \Xi^{\mathsf{lin}}_{\hat{n}_0+\hat{n}_1-\overline{n},n_0+n_1-\overline{n},r_0+r_1}$	$\Xi^{\mathsf{lin}}_{\hat{n}_0,n_0,r_0}\times\Xi^{\mathsf{lin}}_{\hat{n}_1,n_1,r_1} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{\hat{n}_0+\hat{n}_1-\overline{n},n_0+n_1-\overline{n},r_0+r_1}$	$((\hat{n}_0-\overline{n})\cdot r_1+(\hat{n}_1-\overline{n})\cdot r_0)\log\lvert\mathcal{R}_q\rvert$
$\Pi^{\otimes\mathsf{RP}}$	$\Xi^{\mathsf{lin}\text{-}\otimes(r,m')}_{m,\hat{n},r,\beta} \mapsto \Xi^{\mathsf{lin}\text{-}\otimes(r,m')}_{m,\hat{n}+1,r,\beta}\times\Xi^{\mathsf{lin}\text{-}\otimes(r,m')}_{m,\overline{n}+1,1,\tilde{\beta}}$	$\Xi^{\mathsf{lin}\text{-}\otimes(r,m')}_{m,\hat{n},r,\tilde{\beta}'} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}\text{-}\otimes(r,m')}_{m,\hat{n}+1,r,\beta',\varrho}\times\Xi^{\mathsf{lin}\text{-}\otimes(r,m')}_{m',\overline{n}+1,1,\beta''}$	$(\overline{n}+r)\log\lvert\mathcal{R}_q\rvert$
Π^{norm}	$\Xi^{\mathsf{lin}}_{m,\hat{n},r,\beta} \mapsto \Xi^{\mathsf{lin}}_{m,\hat{n}+2,n+2,r,\beta}$	$\Xi^{\mathsf{lin}}_{m,\hat{n},r,\sqrt{r}\beta} \leftarrow\!\shortmid\ \Xi^{\mathsf{lin}}_{m,\hat{n}+2,n+2,r,\beta'}$	$(2d-1)\log_d(m)\log\lvert\mathbb{F}_{q^e}\rvert+3r\log\lvert\mathcal{R}_q\rvert$

Table 3: Summary of RoKs from [KLNO25] together with a new Π^{norm} RoK. For $\Pi^{b\text{-}\mathsf{decomp}}$, we let $\ell=\lceil\log_b(2\beta+1)\rceil$ for some $b\in\mathbb{N}$. For $\Pi^{\mathsf{norm\text{-}klno24}}$, we let $\ell=\left\lceil\log_b(2\tilde{\beta}+1)\right\rceil$ for some $b\in\mathbb{N}$, where $\tilde{\beta}\geq\beta$ is a parameter of Π^{norm} set as in [KLNO25]. In $\Pi^{\otimes\mathsf{RP}}$, we let $\hat{\beta}:=m_{\mathsf{rp}}\cdot\beta$, $\tilde{\beta}':=2\cdot\sqrt{\mathsf{rad}(\mathsf{f})\cdot m_{\mathsf{rp}}}\cdot\beta''$ as in [KLNO25]. For our comparison, we do not need an exact value of $\tilde{\beta}$.

A Extended Preliminaries

A.1 Algebraic Number Theory

Gadget Matrix. Let $b>1$, $k:=\lceil\log_b\beta\rceil+2$, and $\mathbf{g}=[1,b,\ldots,b^{k-1}]^\intercal\in\mathbb{Z}^k$. A gadget matrix $\mathbf{G}_{b,m,k}$ is defined as $\mathbf{G}_{b,m,k}:=\mathbf{I}_m\otimes\mathbf{g}^\intercal\in\mathbb{Z}^{m\times mk}$. We denote the deterministic function that maps $\mathbf{W}$ to $\mathbf{W}'$ by $\mathbf{G}^{-1}_{b,m,k}:\mathcal{R}^{m\times r}\to\mathcal{R}_q^{mk\times r}$, i.e. $\mathbf{G}_{b,m,k}\mathbf{G}^{-1}_{b,m,k}(\mathbf{W})=\mathbf{W}$. If the dimensions (m,k) are clear from the context, we omit them and write $\mathbf{G}_b$ and $\mathbf{G}_b^{-1}$.

Coordinate Ring. Throughout this work, we will use the coordinate ring $(\mathcal{R}_q^r,+,\odot)$ to shorthand the (coordinate-wise) operation on vectors $\mathcal{R}_q^r$. Multiplication is denoted as $\odot$ and addition as $+$. Further, we naturally define a polynomial ring $\mathcal{R}_q^r[x^\mu]$ over $\mathcal{R}_q^r$ with ℓ variables.

Lemma 6 (Adapted Theorem 4.2 from [BCPS18]). *Let $\mathcal{R}_q\cong(\mathbb{F}_{q^e})^{\varphi/e}$, $f\in\mathcal{R}[X_1,\ldots,X_{\mathsf{f}}]$. Then $\mathsf{Pr}\left[(f(r)=0|r\xleftarrow{\$}\mathcal{R}_q^n)\right]\leq\frac{\deg f}{q^e}$, where $\deg f$ is the total degree of f.*

A.2 Computational Assumption

We state an equivalent formulation of the vanishing short integer solution (vSIS) assumption [CLM23], which has a simpler description and nicely aligns with the notation adopted in this work.

Definition 5 (vSIS Assumption (adapted from [CLM23])). *Let $\mathsf{params}=(\mathcal{R},q,n,\mu,\beta)$ be parametrised by λ, where $\mathcal{R}$ is a ring, $q\in\mathbb{N}$ a modulus, $\beta>0$ a norm bound. The $\mathsf{vSIS}_{\mathsf{params}}$ assumption states that, for any PPT adversary $\mathcal{A}$, the advantage function satisfies*

$$\mathsf{Adv}^{\mathsf{vsis}}_{\mathsf{params},\mathcal{A}}(\lambda):=\Pr\left[\begin{array}{l|l}\mathbf{Fw}=\mathbf{0}\bmod q & \mathbf{F}\leftarrow\!\$\ \mathcal{R}_q^{n\times d^{\otimes\mu}}\\ \|\mathbf{w}\|_{\sigma,2}\leq\beta & \mathbf{w}\leftarrow\mathcal{A}(\mathbf{F})\end{array}\right]\leq\mathsf{negl}(\lambda).$$

A.3 Reduction of Knowledge

In [KLNO24], ternary relations $\Xi\subseteq\{0,1\}^*\times\{0,1\}^*\times\{0,1\}^*$ are considered, where a tuple $(\mathsf{pp},\mathsf{stmt},\mathsf{wit})\in\Xi$ consists of public parameters pp, statement stmt, and witness wit. For simplicity, pp is omitted when it is clear from the context. The definition of a reduction of knowledge in [KLNO24] is streamlined in this work.

Definition 6 (Reduction of Knowledge (adapted from [KLNO24])). *Let Ξ_0, Ξ_1 be ternary relations. A* reduction of knowledge (RoK) *Π from Ξ_0 to Ξ_1, short $\Pi\colon\Xi_0\to\Xi_1$, is defined by two PPT algorithms $\Pi=(\mathcal{P},\mathcal{V})$, the prover $\mathcal{P}$, and the verifier $\mathcal{V}$, with the following interface:*

– $\mathcal{P}(\mathsf{pp}, \mathsf{stmt}_0, \mathsf{wit}_0) \to (\mathsf{stmt}_1, \mathsf{wit}_1)$*: Interactively reduce the input statement* $(\mathsf{pp}, \mathsf{stmt}_0, \mathsf{wit}_0) \in \Xi_0$ *to a new statement* $(\mathsf{pp}, \mathsf{stmt}_1, \mathsf{wit}_1) \in \Xi_1$ *or* $\perp$.
– $\mathcal{V}(\mathsf{pp}, \mathsf{stmt}_0) \to \mathsf{stmt}_1$*: Interactively reduce the task of checking the input statement* $(\mathsf{pp}, \mathsf{stmt}_0)$ *w.r.t* Ξ_0 *to checking a new statement* $(\mathsf{pp}, \mathsf{stmt}_1)$ *w.r.t.* Ξ_1.

A RoK Π is *correct*, if for any honest protocol run (with correct inputs), the prover outputs a witness for the reduced statement (which the verifier outputs). A RoK Π is *knowledge sound* from Ξ_0^{KS} to Ξ_1^{KS} with knowledge error $\kappa(\mathsf{pp}, \mathsf{stmt})$ if there is a *black-box* expected polynomial-time extractor $\mathcal{E}$, which succeeds with probability $\epsilon - \kappa(\mathsf{pp}, \mathsf{stmt})$ if the malicious prover outputs a valid witness for the reduced statement with probability ϵ (on verifier's input $(\mathsf{pp}, \mathsf{stmt})$).

Definition 7 (Knowledge Soundness). *An reduction of knowledge* $\Pi = (\mathcal{P}*, \mathcal{V})$ *from* Ξ_0 *to* Ξ_1 *is* knowledge sound *with* knowledge error $\kappa : \mathbb{N} \to [0, 1]$ *if there exists a knowledge extractor* $\mathcal{E}$*, such that for every statement* $x \in \Xi_0$ *and any prover* $\mathcal{P}^*$*, the extractor* $\mathcal{E}^{\mathcal{P}^*}(x)$ *runs in expected time polynomial in* $|x|$ *(counting calls to* $\mathcal{P}^*$ *as unit-cost operations) and outputs a witness* w *such that*

$$\mathsf{Pr}\left[(x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \Xi_0\right] \geq \epsilon(\mathcal{P}^*, x) - \kappa(|x|),$$

where $\epsilon(\mathcal{P}^*, x) \coloneqq \mathsf{Pr}\left[\Xi_1 \leftarrow (\mathcal{P}^*, \mathcal{V})(x)\right]$.

We remark that our definition of knowledge soundness is stronger than the one in [ACK21], as the probability of the extractor's success is defined without a denominator $q(|x|)$ for a polynomial q.

B Variants of Sumcheck Relations

In this section we formalise the variants $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ of the sumcheck relation and the reduction between $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ and the LDE relations $\Xi^{\mathsf{lde}(\text{-}\otimes)}$. These components allow us to describe the RoKs for an alternative batching protocol introduced in Section 6.3 and the RoK from the committed $\mathtt{R1CS}$ relation $\Xi^{\mathsf{lin\text{-}r1cs}}$ defined in Appendix C to the principal relation Ξ^{lin}.

B.1 Π^{sum}: RoK from $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ to $\Xi^{\mathsf{lde}(\text{-}\otimes)}$

Similarly to the RoK $\Pi^{\overline{\mathsf{sum}}}$, we adapt the sumcheck protocol over a finite field to the $\mathcal{R}_q$ setting, and cast it as a RoK from the sumcheck relation Ξ^{sum} to the low-degree extension relation Ξ^{lde}. We also define another variant $\Xi^{\mathsf{sum}\text{-}\otimes}$ of the sumcheck relation. Relations $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ combine multiple relations $((\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i), \mathbf{W}_i)_{i \in \ell}$ and concerns a function $f \in \mathcal{R}_q^r[\mathsf{x}^{\tilde{\mu}}]$, which is a polynomial in $\tilde{\mu}$ variables of individual degree d. The function f is defined as $f \coloneqq h(f_{i,j} : (i,j) \in T)$ where $T \coloneqq \{(i,j) : i \in [\ell], j \in [t_i]\}$, $h \in (\mathcal{R}_q^r[\mathsf{x}^{\tilde{\mu}}])[\mathsf{y}^t]$ is a polynomial and $f_{i,j} \in \mathcal{R}_q^r[\mathsf{x}^{\tilde{\mu}}]$ is a low-degree extension of the matrix $\mathbf{M}_{i,j}\mathbf{W}_i$. The relations $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ state that $\sum_{\mathbf{z} \in [d]^{\tilde{\mu}}} f(\mathbf{z}) = \mathbf{s}$. Below, we present the definition of the relations $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ and the reduction Π^{sum} from $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ to $\Xi^{\mathsf{lde}(\text{-}\otimes)}$.

Definition 8 (Ξ^{sum}, $\Xi^{\mathsf{sum}\text{-}\otimes}$: Sumcheck Relations). *For* $i \in [\ell]$*, let* $\Xi_{(i)}^{\mathsf{lin}} = \Xi_{\hat{n}_i, n_i, \mu_i, r, \beta_i}^{\mathsf{lin}}$ *for some common* r*. The relation* Ξ^{sum} *is defined as*

$$\Xi_{\tilde{\mu}}^{\mathsf{sum}}\left[(\Xi_{(i)}^{\mathsf{lin}}, t_i)_{i \in [\ell]}\right] \coloneqq \left\{ \begin{array}{l} ((h, \mathbf{s}, (\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i, (\mathbf{M}_{i,j})_{j \in [t_i]})_{i \in [\ell]}), (\mathbf{W}_i)_{i \in [\ell]}) : \\ \quad ((\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i), \mathbf{W}_i) \in \Xi_{(i)}^{\mathsf{lin}} \\ \quad h \in (\mathcal{R}_q^r[\mathsf{x}^{\tilde{\mu}}])[\mathsf{y}^t]; \quad \mathbf{M}_{i,j} \in \mathcal{R}_q^{\tilde{m} \times m_i}; \ \mathbf{s} \in \mathcal{R}_q^r \\ \quad \sum_{\mathbf{z} \in [d]^{\tilde{\mu}}} h(\mathsf{LDE}[\mathbf{M}_{i,j}\mathbf{W}_i] : i \in [\ell], j \in [t_i])(\mathbf{z}) = \mathbf{s} \bmod q \end{array} \right\}.$$

The structured variant $\Xi^{\mathsf{sum}\text{-}\otimes}$ *is identical to* Ξ^{sum}*, except that the matrices* $\mathbf{M}_{i,j}$ *in a statement are required to be in* $\mathcal{R}_q^{d^{\otimes \tilde{\mu}} \times d^{\otimes \mu}}$.

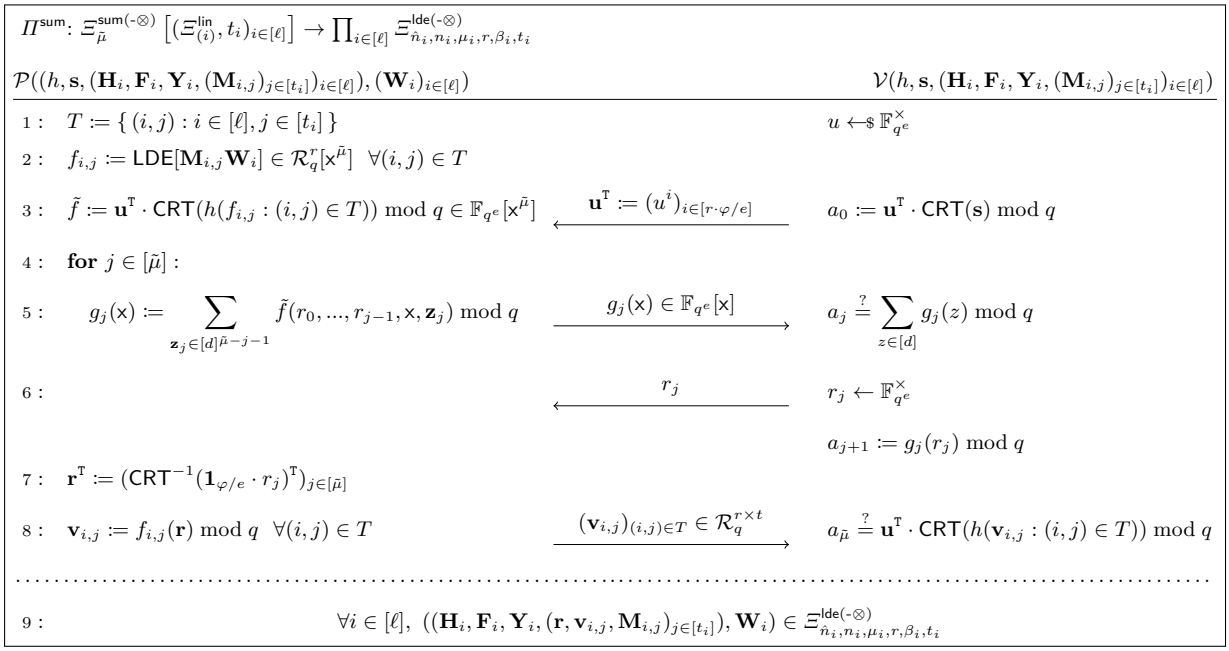

Fig. 4: Π^{sum}: RoK from $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ to $\Xi^{\mathsf{lde}(\text{-}\otimes)}$.

Lemma 7. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, d, \hat{h}, \mu \in \mathbb{N}$, $0 \le \beta_i \le \beta_i' \le q \ \forall i \in [\ell]$ and $h \in (\mathcal{R}_q^r[\mathsf{x}^{\tilde{\mu}}])[\mathsf{y}^t]$ be a polynomial of total degree h' and individual degrees bounded by $\hat{h}(d-1)$. Π^{sum} is a perfectly correct reduction of knowledge for:*

$$\Xi_{\tilde{\mu}}^{\mathsf{sum}(\text{-}\otimes)}\left[(\Xi_{(i)}^{\mathsf{lin}}, t_i)_{i\in[\ell]}\right] \to \prod_{i\in[\ell]} \Xi_{\hat{n}_i,n_i,\mu_i,r,\beta_i,t_i}^{\mathsf{lde}(\text{-}\otimes)}$$

It is knowledge sound with knowledge-error $\kappa = \frac{r\varphi/e-1+\tilde{\mu}\hat{h}(d-1)}{q^e}$ for

$$\Xi_{\tilde{\mu}}^{\mathsf{sum}(\text{-}\otimes)}\left[(\Xi_{(i)}^{\mathsf{lin}}, t_i)_{i\in[\ell]}\right]\left(\cup\Xi_{2\beta_i'}^{\mathsf{sis}}\right)_{i\in[\ell]} \leftarrow \prod_{i\in[\ell]} \Xi_{\hat{n}_i,n_i,\mu_i,r,\beta_i',t_i}^{\mathsf{lde}(\text{-}\otimes)}$$

Let $t \coloneqq \sum_{i\in[\ell]} t_i$. The communication cost is $(\hat{h}(d-1)+1)\tilde{\mu}e\log q + tr\log|\mathcal{R}_q|$. Prover runs in $\mathcal{O}((tm + h')\tilde{m}r)$ ring operations.

Proof. Correctness holds immediately by similarity with the proof of Lemma 3.
Knowledge Soundness: Without loss of generality, suppose $\mathcal{P}^*$ is deterministic and succeeds with a probability ϵ. $\mathcal{E}$ runs $\mathcal{P}^*$ on a random challenge $(u, \mathbf{r})$. If the prover fails, $\mathcal{E}$ aborts. On success, $\mathcal{E}$ obtains $((\mathbf{W}_i)_{i\in[\ell]}, (g_j)_{j\in[\tilde{\mu}]}, (\mathbf{v}_{i,j})_{(i,j)\in T})$ satisfying:

$$\begin{aligned}
&(((\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i), \mathbf{W}_i) \in \Xi^{\mathsf{lin}})_{i\in[\ell]}, \quad \|\mathbf{W}_i\|_{\sigma,2} \le \beta_i',\\
&f_{i,j}(\mathbf{r}) = \mathbf{v}_{i,j} \bmod q,\\
&\mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{s}) = \sum_{z\in[d]} g_0(z) \bmod q\\
&g_j(r_j) = \sum_{z\in[d]} g_{j+1}(z) \bmod q \ \ \forall\ j \in [\tilde{\mu}-1]\\
&a_{\tilde{\mu}} = g_{\tilde{\mu}-1}(r_{\tilde{\mu}-1}) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(h(\mathbf{v}_{i,j} : (i,j) \in T)) \bmod q.
\end{aligned}$$

If $(\mathbf{W}_i)_{i\in[\ell]}$ satisfies $\Xi^{\mathsf{sum}(\text{-}\otimes)}$, the extractor outputs $(\mathbf{W}_i)_{i\in[\ell]}$ and terminates. Otherwise, it re-runs $\mathcal{P}^*$ on fresh challenges $(u', \mathbf{r}')$ until it obtains a second accepting transcript $((\mathbf{W}_i')_{i\in[\ell]}, (g_j')_{j\in[\tilde{\mu}]}, (\mathbf{v}_{i,j}')_{(i,j)\in T})$ satisfying

$$
\begin{aligned}
&(((\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i), \mathbf{W}'_i) \in \Xi^{\mathsf{lin}})_{i\in[\ell]}, \quad \|\mathbf{W}'_i\|_{\sigma,2} \leq \beta'_i,\\
&f_{i,j}(\mathbf{r}') = \mathbf{v}'_{i,j} \bmod q,\\
&\mathbf{u}'^{\mathsf{T}}\mathsf{CRT}(\mathbf{s}) = \sum_{z\in[d]} g'_0(z) \bmod q\\
&g'_j(r'_j) = \sum_{z\in[d]} g'_{j+1}(z) \bmod q \ \ \forall\ j \in [\tilde{\mu}-1]\\
&a'_{\tilde{\mu}} = g'_{\tilde{\mu}-1}(r'_{\tilde{\mu}-1}) = \mathbf{u}'^{\mathsf{T}} \cdot \mathsf{CRT}(h(\mathbf{v}'_{i,j} : (i,j) \in T)) \bmod q.
\end{aligned}
$$

If $\mathbf{W}'_i \neq \mathbf{W}_i$ for any i, then any non-zero column of $\mathbf{W}'_i - \mathbf{W}_i$ is a solution $\mathbf{w}_i$ to $\overline{\mathbf{F}}_i\mathbf{w}_i = \mathbf{0} \bmod q$, and hence $(\overline{\mathbf{F}}_i, \mathbf{w}_i) \in \Xi^{\mathsf{vsis}}_{2\beta'_i}$.

As we suppose that $((h, \mathbf{s}, (\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i, (\mathbf{M}_{i,j})_{j\in[t_i]})_{i\in[\ell]}), (\mathbf{W}_i)_{i\in[\ell]}) \notin \Xi^{\mathsf{sum}(\text{-}\otimes)}$, we show that, under the conditions specified above, this happens with probability κ/ϵ.

We have that since $f_{i,j}(\mathbf{r}') = \mathbf{v}'_{i,j} \bmod q$ and $\tilde{f} \coloneqq \mathbf{u}'^{\mathsf{T}} \cdot h(f_{i,j} : (i,j) \in T) \bmod q \in \mathcal{R}_q[\mathsf{x}^{\tilde{\mu}}]$, then $\tilde{f}(\mathbf{r}') = a'_{\tilde{\mu}}$. The proof is by induction on $\tilde{\mu}$. For $\tilde{\mu} = 1$, the prover's only message specifies a degree $\hat{h}(d-1)$ univariate polynomial $g'_0(\mathsf{x})$. If $g'_0(\mathsf{x}) \neq \tilde{f}(\mathsf{x}) \bmod q$, then because any two distinct degree-$\hat{h}(d-1)$ univariate polynomials over finite $\mathbb{F}_{q^e}$ field can agree at most $\hat{h}(d-1)$ inputs, $g'_0(r'_0) = \tilde{f}_i(r'_0) \bmod q$ with probability at most $\hat{h}(d-1)/(\epsilon q^e)$ over the choice of $r'_{i,0}$. By induction and union bound, for all $\tilde{\mu}$ univariate polynomials, the sumcheck protocol has knowledge soundness error at most κ_0/ϵ, where $\kappa_0 \coloneqq \tilde{\mu}\hat{h}(d-1)/q^e$.

As established, except for the probability κ_0/ϵ, $\sum_{\mathbf{z}\in[d]^{\tilde{\mu}}} \tilde{f}(\mathbf{z}) = a'_{\tilde{\mu}} \bmod q$.

Thus, since now we assume:

$$
\mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(h(f_{i,j} : (i,j) \in T)(\mathbf{z})) = \mathbf{u}^{\mathsf{T}} \cdot \mathsf{CRT}(\mathbf{s}) \bmod q. \tag{2}
$$

If it does not hold that $h(f_{i,j} : (i,j) \in T)(\mathbf{z}) = \mathbf{s}$, Eq. (2) would fail except for the probability at most κ_1/ϵ, where $\kappa_1 \coloneqq (r\varphi/e - 1)/q^e$ by the Schwartz-Zippel lemma over $\mathbb{F}_{q^e}$. Assuming now that $s_k = h(f_{i,j} : (i,j) \in T)(\mathbf{z}) \bmod q$, we yield a contradiction with a supposition that

$((h, \mathbf{s}, (\mathbf{H}_i, \mathbf{F}_i, \mathbf{Y}_i, (\mathbf{M}_{i,j})_{j\in[t_i]})_{i\in[\ell]}), (\mathbf{W}_i)_{i\in[\ell]}) \notin \Xi^{\mathsf{sum}(\text{-}\otimes)}_{\tilde{\mu}} \left[(\Xi^{\mathsf{lin}}_{(i)}, t_i)_{i\in[\ell]}\right]$.

We conclude that $\mathcal{E}$ outputs either:

- $\mathbf{W}$ satisfying the $\Xi^{\mathsf{sum}(\text{-}\otimes)}_{\tilde{\mu}} \left[(\Xi^{\mathsf{lin}}_{(i)}, t_i)_{i\in[\ell]}\right]$ relation, or
- $\mathbf{w}_i$ satisfying $(\overline{\mathbf{F}}_i, \mathbf{w}_i) \in \Xi^{\mathsf{vsis}}_{2\beta'_i}$

with a probability

$$
\epsilon(1 - \frac{\kappa_0 + \kappa_1}{\epsilon}) \leq \epsilon - \frac{\tilde{\mu}\hat{h}(d-1) + r\varphi/e - 1}{q^e} = \epsilon - \kappa.
$$

Expected runtime of $\mathcal{E}$. We argue that Π^{sum} admits an expected polynomial-time knowledge extractor when the prover $\mathcal{P}^*$ succeeds with probability ϵ. The argument follows the same reasoning as in Lemma 3 and requires no further elaboration.

Runtime and Communication Cost: Computing $(f_{i,j})_{(i,j)\in T}$ could be performed in $\mathcal{O}(t\tilde{m}mr)$ ring operations, accounting for matrix multiplication. Further, computing $\tilde{f}$ is performed in $\mathcal{O}(h'\tilde{m}r)$ time, employing regular Horner's method. The sumcheck part is optimised using dynamic programming techniques that store and update partially evaluated polynomials across rounds. These optimisations, described in Lemma 3, reduce the total cost to linear in $\mathcal{O}(\tilde{m})$ by observing a geometric decay in per-round complexity. Therefore, the total time complexity is $\mathcal{O}((tm + h')\tilde{m}r \log|\mathcal{R}_q|)$.

The communication includes sending degree $h'(d-1)$ polynomials $(g_i)_{i\in[\tilde{\mu}]}$ represented by $h'(d-1)+1$ coefficients, and t vectors of length r. In total, the communication cost is $(\hat{h}(d-1)+1)\tilde{\mu}e \log q + tr \log|\mathcal{R}_q|$. □

B.2 Π^{lde}: RoK from Ξ^{lde} to $\Xi^{\mathsf{sum}\text{-}\otimes}$

The following lemma outlines the reduction of the unstructured LDE relation Ξ^{lde} to the structured sumcheck relation $\Xi^{\mathsf{sum}\text{-}\otimes}$. The algorithm restores the structure of the relation by effectively considering the unstructured components of the relation as a part of the witness. This approach requires the verifier to validate the consistency of the "extended witness". A key insight is that this consistency check can be performed during the preprocessing phase.

$$\Pi^{\mathsf{lde}} : \Xi^{\mathsf{lde}}_{\hat{n},n,\mu,r,\beta,t} \to \Xi^{\mathsf{sum}\text{-}\otimes}_{\mu+\tilde{\mu}+\mu_t} \begin{bmatrix} \Xi^{\mathsf{lin}}_{\overline{n},n,\mu,r,\beta}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},\mu+\tilde{\mu}+\mu_q+\mu_t,r,\tilde{\beta}}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n}+t,\tilde{\mu}+\mu_q+\mu_t,r,\hat{\beta}} \end{bmatrix}$$

$\mathcal{P}((\mathbf{H},\mathbf{F},\mathbf{Y},(\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]}),\mathbf{W})$		$\mathcal{V}(\mathbf{H},\mathbf{F},\mathbf{Y},(\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]})$
Preprocessing:		$\widetilde{\mathbf{F}} \leftarrow\!\$\ \mathcal{R}_q^{\overline{n}\times d^{\otimes(\mu+\tilde{\mu}+\mu_t+\mu_q)}}$
	$\xleftarrow{\widetilde{\mathbf{F}},\widehat{\mathbf{F}}}$	$\widehat{\mathbf{F}} \leftarrow\!\$\ \mathcal{R}_q^{\overline{n}\times d^{\otimes(\tilde{\mu}+\mu_t+\mu_q)}}$
		$\mathbf{U} \coloneqq \mathbf{G}_b^{-1}\left(\mathbf{1}_r^{\mathsf{T}} \otimes \mathsf{vec}\left((\mathbf{M}_i^{\mathsf{T}})_{i\in[t]}\right)\right) \in \mathcal{R}_q^{d^{\mu+\tilde{\mu}+\mu_t+\mu_q}\times r}$
		$\widetilde{\mathbf{Y}} = \widetilde{\mathbf{F}}\cdot\mathbf{U} \bmod q \in \mathcal{R}_q^{\overline{n}\times r}$
Online:		
$\mathbf{V}^{\mathsf{T}} \coloneqq \left((\mathbf{M}_i\mathbf{W})^{\mathsf{T}}\right)_{i\in[t]}$		
$\widehat{\mathbf{V}} \coloneqq \mathbf{G}_b^{-1}(\mathbf{V}) \in \mathcal{R}^{d^{\tilde{\mu}+\mu_t+\mu_q}\times r}$		
$\widehat{\mathbf{Y}} \coloneqq \widehat{\mathbf{F}}\widehat{\mathbf{V}} \bmod q$	$\xrightarrow{\widehat{\mathbf{Y}}}$	
	$\xleftarrow{\mathbf{c}}$	$c \leftarrow\!\$\ \mathcal{C}_{\mathcal{R}_q}\quad \mathbf{c}^{\mathsf{T}} \coloneqq (1,c,\ldots,c^{t\tilde{m}-1})$

$$h(\mathsf{y}_w,\mathsf{y}_u,\mathsf{y}_v) \coloneqq (\mathsf{y}_w \odot \mathsf{y}_u - \mathsf{y}_v)\cdot \mathsf{LDE}[\mathbf{c}\otimes\mathbf{1}_m]$$

$$\mathbf{M}_w \coloneqq \mathbf{1}_{\tilde{m}t}\otimes\mathbf{I}_m \quad \mathbf{M}_u \coloneqq \mathbf{G}_b \quad \mathbf{M}_v \coloneqq (\mathbf{I}_{\tilde{m}t}\otimes\mathbf{e}_{0,m})\mathbf{G}_b$$

$$\mathbf{S}^{\mathsf{T}} \coloneqq (\mathbf{s}_i)_{i\in[t]} \quad \mathbf{E} \coloneqq ((\mathbf{e}_{i,t}\otimes\widetilde{\mathbf{r}}_i)_{i\in[t]})^{\mathsf{T}}\mathbf{G}_b \quad /\!/\ \mathsf{LDE}[\mathbf{W}](\mathbf{r})^{\mathsf{T}} = \widetilde{\mathbf{r}}_i^{\mathsf{T}}\mathbf{W}$$

$$\left(\left(h,\ \mathbf{0}_{\mathbf{r}},\ \begin{pmatrix} \mathbf{H}, & \mathbf{F}, & \mathbf{Y}, & \mathbf{M}_w \\ \mathbf{I}_{\overline{n}}, & \widetilde{\mathbf{F}}, & \widetilde{\mathbf{Y}}, & \mathbf{M}_u \\ \mathbf{I}_{\overline{n}+t}, & \begin{bmatrix}\widehat{\mathbf{F}}\\ \mathbf{E}\mathbf{G}_b\end{bmatrix}, & \begin{bmatrix}\widehat{\mathbf{Y}}\\ \mathbf{S}\end{bmatrix}, & \mathbf{M}_v \end{pmatrix}\right),\ \begin{pmatrix}\mathbf{W}\\ \mathbf{U}\\ \widehat{\mathbf{V}}\end{pmatrix}\right) \in \Xi^{\mathsf{sum}\text{-}\otimes}_{\mu+\tilde{\mu}+\mu_t} \begin{bmatrix} \Xi^{\mathsf{lin}}_{\overline{n},n,\mu,r,\beta}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},\mu+\tilde{\mu}+\mu_q+\mu_t,r,\tilde{\beta}}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n}+t,\tilde{\mu}+\mu_q+\mu_t,r,\hat{\beta}} \end{bmatrix}$$

Fig. 5: Π^{lde}: A reduction from Ξ^{lde} to $\Xi^{\mathsf{sum}\text{-}\otimes}$

Lemma 8. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, d, t, \mu, \tilde{\mu} \in \mathbb{N}$, $0 \le \beta \le \beta' \le q$, $0 \le \tilde{\beta} \le \tilde{\beta}' \le q$, $0 \le \hat{\beta} \le \hat{\beta}' \le q$, $\tilde{m} = d^{\tilde{\mu}}$, $m = d^{\mu}$, $t = d^{\mu_t}$, b is set such that $\|\mathbf{G}_b^{-1}(\mathbf{m})\|_{\sigma,2} \le \tilde{\beta}$ for any $\mathbf{m} \in \mathcal{R}_q^{d^{\mu+\tilde{\mu}+\mu_t}}$ and $\|\mathbf{G}_b^{-1}(\mathbf{n})\|_{\sigma,2} \le \hat{\beta}$ for any $\mathbf{n} \in \mathcal{R}_q^{d^{\tilde{\mu}+\mu_t}}$. Further, let $\gamma = \log_b q$ and $\gamma = d^{\mu_q}$.*

Π^{lde} is a perfectly correct reduction of knowledge for:

$$\Xi^{\mathsf{lde}}_{\hat{n},n,\mu,r,\beta,t} \to \Xi^{\mathsf{sum}\text{-}\otimes}_{\mu+\tilde{\mu}+\mu_t} \begin{bmatrix} \Xi^{\mathsf{lin}}_{\overline{n},n,\mu,r,\beta}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},\mu+\tilde{\mu}+\mu_q+\mu_t,r,\tilde{\beta}}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n}+t,\tilde{\mu}+\mu_q+\mu_t,r,\hat{\beta}} \end{bmatrix}.$$

It is knowledge sound with knowledge-error $\kappa = {}^{\tilde{m}t-1}/_{q^e}$ *for*

$$\Xi^{\mathsf{lde}}_{\hat{n},n,\mu,r,\beta',t} \cup \Xi^{\mathsf{sis}}_{2\beta'} \cup \Xi^{\mathsf{sis}}_{2\tilde{\beta}'} \cup \Xi^{\mathsf{sis}}_{2\hat{\beta}'} \leftarrow \Xi^{\mathsf{sum}\text{-}\otimes}_{\mu+\tilde{\mu}+\mu_t} \begin{bmatrix} \Xi^{\mathsf{lin}}_{\overline{n},n,\mu,r,\beta'}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},\mu+\tilde{\mu}+\mu_q+\mu_t,r,\tilde{\beta}'}, \\ \Xi^{\mathsf{lin}}_{\overline{n},\overline{n}+t,\tilde{\mu}+\mu_q+\mu_t,r,\hat{\beta}'} \end{bmatrix}.$$

The prover's communication cost is $\overline{n}r \log |\mathcal{R}_q|$ *and the prover runs in* $\mathcal{O}(m\tilde{m}rt\overline{n})$ *ring operations. The verifier runs in* $\mathcal{O}(m\tilde{m}rt\overline{n})$ *ring operations during preprocessing and has constant runtime in the online phase.*

Proof. Correctness: We consider an instance $((\mathbf{H}, \mathbf{F}, \mathbf{Y}, (\mathbf{r}_i, \mathbf{s}_i, \mathbf{M}_i)_{i\in[t]}), \mathbf{W}) \in \Xi^{\mathsf{lde}}$. We write $f_{m_{i,j}} = \mathsf{LDE}[\mathbf{m}_{i,j}]$ for the low-degree extension based on the vector $\mathbf{m}_{i,j}$ defined as an j-th row of $\mathbf{M}_i$. Let $(\mathbf{v}_k)_{k\in[r]} \coloneqq \mathbf{V}$, $(v_{i,j,k})_{(i,j)\in[t,\tilde{m}]} \coloneqq \mathbf{v}_k^{\mathsf{T}}$ and $(\mathbf{w}_k)_{k\in[r]}$. From that we have immediately $v_{i,j,k} = \langle \mathbf{m}_{i,j}, \mathbf{w}_k \rangle$ for all $k \in [r]$. This immediately translates to

$$\forall\ (i,j,k) \in [t,\tilde{m},r] \quad \sum_{\mathbf{z}\in[d]^\mu} (\mathsf{LDE}[\mathbf{m}_{i,j}] \cdot \mathsf{LDE}[\mathbf{w}_k])(\mathbf{z}) - v_{i,j,k} = 0 \bmod q.$$

or

$$\forall\ (i,j,k) \in [t,\tilde{m},r] \quad \sum_{\mathbf{z}\in[d]^\mu} (\mathsf{LDE}[\mathbf{m}_{i,j}] \cdot \mathsf{LDE}[\mathbf{w}_k] - \mathsf{LDE}[\mathbf{e}_{0,m} \cdot v_{i,j,k}])(\mathbf{z}) = 0 \bmod q.$$

Upon batching all instances for $(i,j) \in [t,\tilde{m}]$ with a challenge vector $\mathbf{c}$ and extending to a multi-column case, we have:

$$\sum_{\mathbf{z}\in[d]^{\mu+\tilde{\mu}+\mu_t}} (\mathsf{LDE}[\mathbf{c}\otimes\mathbf{1}_m](\mathsf{LDE}[\mathbf{1}_r^{\mathsf{T}} \otimes \mathsf{vec}\left((\mathbf{M}_i^{\mathsf{T}})_{i\in[t]}\right)] \odot \mathsf{LDE}[\mathbf{1}_{\tilde{m}t} \otimes \mathbf{W}] - \mathsf{LDE}[(\mathbf{I}_{\tilde{m}t} \otimes \mathbf{e}_{0,m})\mathbf{V}]))(\mathbf{z}) = \mathbf{0}_r \bmod q.$$

Eventually, we consider $\mathbf{G}$-decomposition explicitly:

$$\sum_{\mathbf{z}\in[d]^{\mu+\tilde{\mu}+\mu_t}} (\mathsf{LDE}[\mathbf{c}\otimes\mathbf{1}_m](\mathsf{LDE}[\mathbf{G}_b\mathbf{U}] \odot \mathsf{LDE}[(\mathbf{1}_{\tilde{m}t} \otimes \mathbf{I}_m)\mathbf{W}] - \mathsf{LDE}[(\mathbf{I}_{\tilde{m}t} \otimes \mathbf{e}_{0,m})\mathbf{G}_b\widehat{\mathbf{V}}]))(\mathbf{z}) = \mathbf{0}_r \bmod q. \tag{3}$$

We observe that since $(\mathbf{e}_{i,t} \otimes \widetilde{\mathbf{r}}_i)^{\mathsf{T}} \cdot \mathbf{G}_b\widehat{\mathbf{V}} = \widetilde{\mathbf{r}}_i^{\mathsf{T}} \cdot \mathbf{V} \bmod q$ and further $\mathbf{V}^{\mathsf{T}} \coloneqq \left((\mathbf{M}_i\mathbf{W})^{\mathsf{T}}\right)_{i\in[t]}$, then $(\mathbf{e}_{i,t} \otimes \widetilde{\mathbf{r}}_i)^{\mathsf{T}} \cdot \mathbf{G}_b\widehat{\mathbf{V}} = \mathbf{s}_i^{\mathsf{T}} \bmod q$ as expected. By substituting $\mathbf{M}_w$ with $\mathbf{1}_{\tilde{m}t} \otimes \mathbf{I}_m$, $\mathbf{G}_b$ with $\mathbf{M}_u$ and $(\mathbf{I}_{\tilde{m}t} \otimes \mathbf{e}_{0,m})\mathbf{G}_b$ with $\mathbf{M}_v$ in Eq. (3), we show that indeed the output satisfies $\Xi^{\mathsf{sum}\text{-}\otimes}_{\tilde{\mu}}\left[(\Xi^{\mathsf{lin}}_{(i)}, t_i)_{i\in[\ell]}\right]$ relation. The norm bounds are satisfied due to the selection of the decomposition base b.

Runtime and Communication cost: We observe that the prover runtime requires only linear operations on the witness and is upper-bounded by $\mathcal{O}(m\tilde{m}rt\overline{n})$ operations on ring elements. For the verifier, we consider two phases. The online phrase requires $\mathcal{O}(1)$ operations. The preprocesses phase requires $\mathcal{O}(m\tilde{m}rt\overline{n})$ ring elements operations.

The prover's communication cost include sending $\widehat{\mathbf{Y}}$, i.e. $\overline{n} \cdot r$ ring elements.

Knowledge Soundness: Without loss of generality, assume that the prover is deterministic and succeeds with a probability ϵ. Consider a knowledge extractor $\mathcal{E}$ which runs the prover $\mathcal{P}^*$ on some challenge $\mathbf{c}$. If the prover fails, $\mathcal{E}$ aborts. On success, it obtains $(\mathbf{W}, \mathbf{U}, \widehat{\mathbf{V}})$ such that:

$$\begin{aligned} \mathbf{HFW} &= \mathbf{Y} \bmod q, & \|\mathbf{W}\|_{\sigma,2} &\le \beta, \\ \widetilde{\mathbf{F}}\mathbf{U} &= \widetilde{\mathbf{Y}} \bmod q, & \|\mathbf{U}\|_{\sigma,2} &\le \tilde{\beta}, \\ \widehat{\mathbf{F}}\widehat{\mathbf{V}} &= \widehat{\mathbf{Y}} \bmod q, & \|\widehat{\mathbf{V}}\|_{\sigma,2} &\le \hat{\beta}, \end{aligned}$$

$$(\mathbf{e}_{i,t} \otimes \widetilde{\mathbf{r}}_i)^{\mathsf{T}}\widehat{\mathbf{V}} = \mathbf{s}_i^{\mathsf{T}} \bmod q \ \ \forall\ i \in [t],$$

and

$$\sum_{\mathbf{z}\in[d]^{\mu+\tilde{\mu}+\mu_t}} (\mathsf{LDE}[\mathbf{c}\otimes\mathbf{1}_m](\mathsf{LDE}[\mathbf{G}_b\mathbf{U}]\odot\mathsf{LDE}[\mathbf{1}_{\tilde{m}t}\otimes\mathbf{W}] - \mathsf{LDE}[(\mathbf{I}_{\tilde{m}t}\otimes\mathbf{e}_{0,m})\mathbf{G}_b\widehat{\mathbf{V}}]))(\mathbf{z}) = \mathbf{0}_r \bmod q.$$

If $((\mathbf{H},\mathbf{F},\mathbf{Y},(\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]}),\mathbf{W})\in\Xi^{\mathsf{lde}}$, the extractor terminates and outputs $\mathbf{W}$ as the witness. Otherwise, it rewinds the prover P^* until it succeeds in convincing the verifier. The extractor $\mathcal{E}$ obtains the second accepting transcript $(\mathbf{W}',\mathbf{U}',\widehat{\mathbf{V}}')$ with a fresh challenge $\mathbf{c}$ such that:

$$\begin{aligned}\mathbf{HFW}' &= \mathbf{Y}' \bmod q, & \|\mathbf{W}'\|_{\sigma,2} &\le \beta,\\ \widetilde{\mathbf{F}}\mathbf{U}' &= \widetilde{\mathbf{Y}}' \bmod q, & \|\mathbf{U}'\|_{\sigma,2} &\le \tilde{\beta},\\ \widehat{\mathbf{F}}\widehat{\mathbf{V}}' &= \widehat{\mathbf{Y}}' \bmod q, & \|\widehat{\mathbf{V}}'\|_{\sigma,2} &\le \hat{\beta},\end{aligned}$$

$$(\mathbf{e}_{i,t}\otimes\widetilde{\mathbf{r}}_i)^{\mathsf{T}}\widehat{\mathbf{V}}' = \mathbf{s}_i^{\mathsf{T}} \bmod q \quad \forall\ i\in[t],$$

and

$$\sum_{\mathbf{z}\in[d]^{\mu+\tilde{\mu}+\mu_t}} (\mathsf{LDE}[\mathbf{c}'\otimes\mathbf{1}_m](\mathsf{LDE}[\mathbf{G}_b\mathbf{U}']\odot\mathsf{LDE}[\mathbf{1}_{\tilde{m}t}\otimes\mathbf{W}'] - \mathsf{LDE}[(\mathbf{I}_{\tilde{m}t}\otimes\mathbf{e}_{0,m})\mathbf{G}_b\widehat{\mathbf{V}}']))(\mathbf{z}) = \mathbf{0}_r \bmod q. \tag{4}$$

The extractor $\mathcal{E}$ checks:

(i) whether $\mathbf{U}\neq\mathbf{1}_r^{\mathsf{T}}\otimes\mathsf{vec}\Big((\mathbf{M}_i^{\mathsf{T}})_{i\in[t]}\Big)$ or $\mathbf{U}'\neq\mathbf{1}_r^{\mathsf{T}}\otimes\mathsf{vec}\Big((\mathbf{M}_i^{\mathsf{T}})_{i\in[t]}\Big)$. If either of them holds, then $\widetilde{\mathbf{F}}\widetilde{\mathbf{m}}=0$ for some column $\widetilde{\mathbf{m}}$ and $(\widetilde{\mathbf{F}},\widetilde{\mathbf{m}})'\in\Xi^{\mathsf{vsis}}_{2\tilde{\beta}'}$ and the extractor outputs $\widetilde{\mathbf{m}}$ and terminates.

(ii) if $\mathbf{W}'\neq\mathbf{W}$. Then, let $\mathbf{w}$ be any non-zero column of $\mathbf{W}'-\mathbf{W}$, and therefore $(\overline{\mathbf{F}},\mathbf{w})\in\Xi^{\mathsf{vsis}}_{2\beta'}$. The extractor outputs $\mathbf{w}$ and terminates.

(iii) if $\widehat{\mathbf{V}}'\neq\widehat{\mathbf{V}}$. Then, let $\mathbf{v}$ be any non-zero column of $\widehat{\mathbf{V}}'-\widehat{\mathbf{V}}$, and therefore $(\widehat{\mathbf{F}},\mathbf{v})\in\Xi^{\mathsf{vsis}}_{2\hat{\beta}'}$. The extractor outputs $\mathbf{v}$ and terminates.

Since now, we turn our focus on the case when $\mathbf{W}=\mathbf{W}'$, $\widehat{\mathbf{V}}=\widehat{\mathbf{V}}'$ and $\mathbf{U}=\mathbf{U}'=\mathbf{1}_r^{\mathsf{T}}\otimes\mathsf{vec}\Big((\mathbf{M}_i^{\mathsf{T}})_{i\in[t]}\Big)$. We recall that $((\mathbf{H},\mathbf{F},\mathbf{Y},(\mathbf{r}_i,\mathbf{s}_i,\mathbf{M}_i)_{i\in[t]}),\mathbf{W})\notin\Xi^{\mathsf{lde}}$. We will show this, conditioned on the aforementioned case, this could happen with probability at most κ/ϵ.

In other words, we assume that for $\mathbf{m}_{i,j}$ and $\mathbf{w}_k$ defined as during the correctness proof, the following holds:

$$\sum_{\mathbf{z}\in[d]^{\mu}} (\mathsf{LDE}[\mathbf{m}_{i,j}]\cdot\mathsf{LDE}[\mathbf{w}_k]-\mathsf{LDE}[\mathbf{e}_{0,m}\cdot v_{i,j,k}])(\mathbf{z})\neq 0 \bmod q$$

for some $(i,j,k)\in[t,\tilde{m},r]$, but after transforming Eq. (4), we have:

$$\sum_{\mathbf{z}\in[d]^{\mu+\tilde{\mu}+\mu_t}} (\mathsf{LDE}[\mathbf{c}'\otimes\mathbf{1}_m](\mathsf{LDE}[\mathbf{1}_r^{\mathsf{T}}\otimes\mathsf{vec}\left((\mathbf{M}_i^{\mathsf{T}})_{i\in[t]}\right)]\odot\mathsf{LDE}[\mathbf{1}_{\tilde{m}t}\otimes\mathbf{W}] - \mathsf{LDE}[(\mathbf{I}_{\tilde{m}t}\otimes\mathbf{e}_{0,m})\mathbf{V}]))(\mathbf{z}) = \mathbf{0}_r \bmod q.$$

In other words, we have:

$$\forall k\in[r]\quad \sum_{(i,j)\in[t,\tilde{m}]} c'_{i\tilde{m}+j} \sum_{\mathbf{z}\in[d]^{\mu}} (\mathsf{LDE}[\mathbf{m}_{i,j}]\cdot\mathsf{LDE}[\mathbf{w}_k]-\mathsf{LDE}[\mathbf{e}_{0,m}\cdot v_{i,j,k}])(\mathbf{z}) = 0 \bmod q,$$

which, since $\mathbf{c}'$ is sampled conditioned only on the prover's acceptance, happens with probability at most κ/ϵ, where $\kappa \coloneqq {}^{\tilde{m}t-1}/_{q^e}$ under the Schwartz-Zippel lemma.

We conclude that $\mathcal{E}$ outputs either:

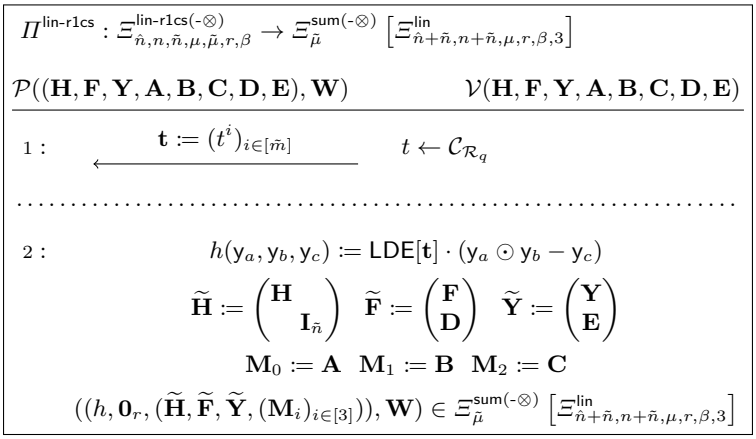

Fig. 6: $\Pi^{\mathsf{lin\text{-}r1cs}}$: RoK from $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ to $\Xi^{\mathsf{sum}(\text{-}\otimes)}$.

- $\mathbf{W}$ satisfying the Ξ^{lde} relation, or
- $\widetilde{\mathbf{m}}$ satisfying the $\Xi^{\mathsf{vsis}}_{2\hat{\beta}'}$ relation, or
- $\mathbf{v}$ satisfying the $\Xi^{\mathsf{vsis}}_{2\hat{\beta}'}$ relation, or
- $\mathbf{w}$ satisfying the $\Xi^{\mathsf{vsis}}_{2\beta'}$ relation

with probability $\epsilon(1-\kappa/\epsilon) = \epsilon - \kappa$.

Expected runtime of $\mathcal{E}$. We argue that Π^{lde} admits an expected polynomial-time knowledge extractor when the prover $\mathcal{P}^*$ succeeds with probability ϵ. The argument follows the same reasoning as in Lemma 3 and requires no further elaboration.

□

Remark 3. Matrices $\mathbf{F}$, $\widehat{\mathbf{F}}$ and $\widetilde{\mathbf{F}}$ could be sampled as consecutive row-tensor extensions of one another, e.g.

$$\widehat{\mathbf{F}} := \widehat{\mathbf{L}} \bullet \mathbf{F} \text{ and } \widetilde{\mathbf{F}} := \widetilde{\mathbf{L}} \bullet \widehat{\mathbf{F}}$$

and a complete folding-based protocol in the style of [KLNO24] can take advantage and combine those relations in the round when the commitment keys agree. In particular, after folding the largest instance (with commitment key $\widetilde{\mathbf{F}}$) down to the size of the instance with commitment key $\widehat{\mathbf{F}}$, the witnesses can be combined by horizontal concatenation of the witness as they will share the commitment key. A similar reasoning can then be applied when reaching the size of the smallest instance corresponding to commitment key $\mathbf{F}$.

C Rank-1 Constraint System

In this section we formalise the committed rank-1 constraint system (R1CS) relations $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ and the RoK from $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ to the principal relation Ξ^{lin}. The `R1CS` relation is widely-used to capture an arithmetic computation. The committed `R1CS` relation is defined as in Section 7.

C.1 $\Pi^{\mathsf{lin\text{-}r1cs}}$: RoK from $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ to $\Xi^{\mathsf{sum}(\text{-}\otimes)}$

The following lemma holds for the RoK $\Pi^{\mathsf{lin\text{-}r1cs}}$ from $\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}$ to $\Xi^{\mathsf{sum}(\text{-}\otimes)}$ provided in Fig. 6.

Lemma 9. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r, \mu, d, \tilde{d} \in \mathbb{N}$ and $0 \le \beta \le \beta' \le q$. $\Pi^{\mathsf{lin\text{-}r1cs}}$ is a perfectly correct reduction of knowledge for:*

$$\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta} \to \Xi^{\mathsf{sum}(\text{-}\otimes)}_{\tilde{\mu}} \left[\Xi^{\mathsf{lin}}_{\hat{n}+\tilde{n},n+\tilde{n},\mu,r,\beta}, 3\right].$$

It is knowledge sound with knowledge-error $(\tilde{m}-1)/q^e$ for

$$\Xi^{\mathsf{lin\text{-}r1cs}(\text{-}\otimes)}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta'} \cup \Xi^{\mathsf{sis}}_{2\beta'} \leftarrow \Xi^{\mathsf{sum}(\text{-}\otimes)}_{\tilde{\mu}} \left[\Xi^{\mathsf{lin}}_{\hat{n}+\tilde{n},n+\tilde{n},\mu,r,\beta}, 3\right].$$

The communication cost is 0 ring elements.

Proof. Correctness: Consider the following instance: $((\mathbf{H},\mathbf{F},\mathbf{Y},\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{E}),\mathbf{W}) \in \Xi^{\mathsf{lin\text{-}r1cs\text{-}}\otimes}_{\hat{n},n,\tilde{n},\mu,\tilde{\mu},r,\beta}$. We recall that $\mathbf{AW} \odot \mathbf{BW} = \mathbf{CW}$, and we have that

$$\forall\ \mathbf{z} \in [d]^{\tilde{\mu}}\ \ \mathsf{LDE}[\mathbf{AW}](\mathbf{z}) \odot \mathsf{LDE}[\mathbf{BW}](\mathbf{z}) - \mathsf{LDE}[\mathbf{CW}](\mathbf{z}) = \mathbf{0}_r \bmod q.$$

Consequently,

$$\sum_{\mathbf{z}\in[d]^{\tilde{\mu}}} \mathsf{LDE}[\mathbf{t}](\mathsf{LDE}[\mathbf{AW}](\mathbf{z}) \odot \mathsf{LDE}[\mathbf{BW}](\mathbf{z}) - \mathsf{LDE}[\mathbf{CW}](\mathbf{z})) = \mathbf{0}_r \bmod q$$

and therefore, $((h,\mathbf{0}_r,(\widetilde{\mathbf{H}},\widetilde{\mathbf{F}},\widetilde{\mathbf{Y}},(\mathbf{M}_i)_{i\in[3]})),\mathbf{W}) \in \Xi^{\mathsf{sum}(\text{-}\otimes)}_{\tilde{\mu}}\left[\Xi^{\mathsf{lin}}_{\hat{n}+\tilde{n},n+\tilde{n},\mu,r,\beta},3\right]$.

Knowledge soundness:
Extractor $\mathcal{E}$. Without loss of generality, suppose $\mathcal{P}^*$ is deterministic and succeeds with a probability ϵ. $\mathcal{E}$ runs $\mathcal{P}^*$ on a random challenge t. If the prover fails, $\mathcal{E}$ aborts. On success, $\mathcal{E}$ obtains $\mathbf{W}$ satisfying the relations:

$$\begin{aligned}&\mathbf{HFW} = \mathbf{Y} \bmod q,\\ &\sum_{\mathbf{z}\in[d]^{\tilde{\mu}}} \mathsf{LDE}[\mathbf{t}](\mathbf{z})(\mathsf{LDE}[\mathbf{AW}](\mathbf{z}) \odot \mathsf{LDE}[\mathbf{BW}](\mathbf{z}) - \mathsf{LDE}[\mathbf{CW}](\mathbf{z})) = \mathbf{0}_r \bmod q,\end{aligned}$$

with $\|\mathbf{W}\|_{\sigma,2} \le \beta'$. If $\mathbf{AW} \odot \mathbf{BW} = \mathbf{CW}$, the extractor outputs $\mathbf{W}$ and concludes the extraction. Otherwise, assume that $\Xi^{\mathsf{lin\text{-}r1cs\text{-}}\otimes}$ does not hold. In this case, the extractor re-runs the prover $\mathcal{P}^*$ on fresh challenges t' (possibly many times), until obtaining $\mathbf{W}'$ satisfying the relations:

$$\begin{aligned}&\mathbf{HF}'\mathbf{W}' = \mathbf{Y}' \bmod q,\\ &\sum_{\mathbf{z}\in[d]^{\tilde{\mu}}} \mathsf{LDE}[\mathbf{t}'](\mathbf{z})(\mathsf{LDE}[\mathbf{AW}'](\mathbf{z}) \odot \mathsf{LDE}[\mathbf{BW}'](\mathbf{z}) - \mathsf{LDE}[\mathbf{CW}'](\mathbf{z})) = \mathbf{0}_r \bmod q,\end{aligned}$$

with $\|\mathbf{W}'\|_{\sigma,2} \le \beta'$. If it turns out that $\mathbf{W}' \neq \mathbf{W}$, then any non-zero column $\mathbf{v}$ of $\mathbf{W}' - \mathbf{W}$ satisfies $\overline{\mathbf{F}}\mathbf{v} = \mathbf{0} \bmod q$, and hence $(\overline{\mathbf{F}},\mathbf{v}) \in \Xi^{\mathsf{vsis}}_{2\beta'}$. Since now we assume that $\sum_{\mathbf{z}\in[d]^{\tilde{\mu}}} \mathsf{LDE}[\mathbf{t}'](\mathbf{z})(\mathsf{LDE}[\mathbf{AW}](\mathbf{z}) \odot \mathsf{LDE}[\mathbf{BW}](\mathbf{z}) - \mathsf{LDE}[\mathbf{CW}](\mathbf{z})) = \mathbf{0}_r \bmod q$ or, equivalently:

$$\mathbf{t}'^{\mathsf{T}}(\mathbf{AW} \odot \mathbf{BW} - \mathbf{CW}) = \mathbf{0}_r \bmod q. \tag{5}$$

Let $\mathbf{A}^{\mathsf{T}} := (\mathbf{a}_j)_{j\in[\tilde{m}]}$, $\mathbf{B}^{\mathsf{T}} := (\mathbf{b}_j)_{j\in[\tilde{m}]}$, and $\mathbf{C}^{\mathsf{T}} := (\mathbf{c}_j)_{j\in[\tilde{m}]}$. Since we assumed $\mathbf{Aw} \odot \mathbf{Bw}_i \neq \mathbf{Cw}_i$, there exists a pair $(i,j) \in [r,\tilde{m}]$ such that $\langle \mathbf{a}_j,\mathbf{w}_i\rangle \cdot \langle \mathbf{b}_j,\mathbf{w}_i\rangle \neq \langle \mathbf{c}_j,\mathbf{w}_i\rangle$. Then, by the Schwartz-Zippel lemma, Eq. (5) occurs with probability at most κ/ϵ, where $\kappa := \tilde{m}-1/q^e$. Otherwise, we would have $\mathbf{AW} \odot \mathbf{BW} = \mathbf{CW} \bmod q$, contradicting the supposition that $((\mathbf{H},\mathbf{F},\mathbf{Y},\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{E}),\mathbf{W}) \notin \Xi^{\mathsf{lin\text{-}r1cs\text{-}}\otimes}_{\tilde{n},\mu,\tilde{\mu},r,\beta'}$.

To summarise, with probability at least $\epsilon - \kappa$, the extractor outputs either: (i) $\mathbf{W}$ satisfying the $\Xi^{\mathsf{lin\text{-}r1cs\text{-}}\otimes}$ relation, or (ii) $\mathbf{v}$ satisfying $(\overline{\mathbf{F}},\mathbf{v}) \in \Xi^{\mathsf{vsis}}_{2\beta'}$.

Expected runtime: We argue that $\Pi^{\mathsf{lin\text{-}r1cs}}$ admits an expected polynomial-time knowledge extractor. The argument follows the same reasoning as in Lemma 3. □

Eventually, we observe that the proof of Lemma 5 follows from the connection of arguments in Lemma 9 and Lemma 7. We note the polynomial h specified in $\Pi^{\mathsf{lin\text{-}r1cs}}$ has an individual degree of $3(d-1)$ in each variable as it computes the square of a degree-$(d-1)$ polynomial and multiplies it by another degree-$(d-1)$ polynomial.

D Improved Batching via Sumcheck

In this section, we describe our improved batching technique that serves as a drop-in replacement for the batching technique of [KLNO24, KLNO25]. To recall, the batching technique of [KLNO24, KLNO25] serves the purpose of aggregating rows of the matrix $\mathbf{F}$. More precisely, matrix $\mathbf{F}$ is divided into 2 pieces:

$$\mathbf{F} := \begin{pmatrix}\overline{\mathbf{F}}\\ \underline{\mathbf{F}}\end{pmatrix}.$$

The top rows $\overline{\mathbf{F}}$ are used for the Ajtai commitment key. Bottom rows $\underline{\mathbf{F}}$ are used for expressing the relation and are (typically) ready to be aggregated. The aggregation proposed in [KLNO24, KLNO25] involves combining the contributions from the bottom rows $\underline{\mathbf{F}} \in \mathcal{R}_q^{\underline{n} \times m}$ via a random vector. This solution, although simple, has a significant drawback. Even though, the aggregation reduces the number of rows for the communication, the runtime (of both proving and verification) still remains proportional to $\underline{n}$. Therefore, particularly in the context of folding scheme, the asymptotic (and concrete) efficiency get degraded as soon as the number of repetitions grows to be superconstant.

We address this issue by proposing a new batching technique that relies on the sumcheck protocol. In particular, we show how to express the relation $\underline{\mathbf{HF}}\mathbf{W} = \underline{\mathbf{Y}} \bmod q$ as a sum over a polynomial evaluation defined over a hypercube. As a result, after applying the sumcheck protocol, the reduced evaluation claim over $(\mathsf{LDE}[\mathbf{c}^\mathsf{T}\underline{\mathbf{HF}}]\mathsf{LDE}[\underline{\mathbf{W}}])(\mathbf{r}) = \mathbf{t}^\mathsf{T}$, which is split into two components: $\mathsf{LDE}[\mathbf{c}^\mathsf{T}\underline{\mathbf{HF}}](\mathbf{r}) = \mathbf{t}_0$ and $\mathsf{LDE}[\underline{\mathbf{W}}](\mathbf{r}) = \mathbf{t}_1$. The first component can be computed efficiently by the verifier. The second component serves as a new (singular) structured claim that replaces the original relation $\underline{\mathbf{HF}}\mathbf{W} = \underline{\mathbf{Y}} \bmod q$. We provide a helper RoK Π^{batch^+}, which is a building block for a batching RoK from Ξ^{lin} to Ξ^{lin}.

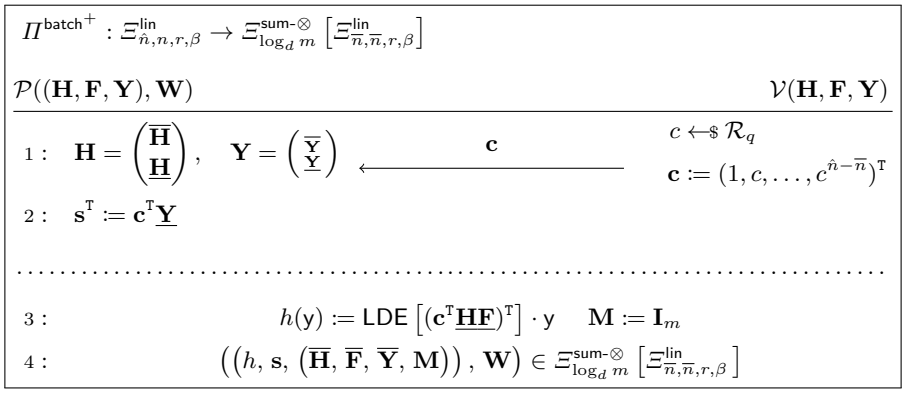

Fig. 7: Π^{batch^+}: A reduction from Ξ^{lin} to $\Xi^{\mathsf{sum}\text{-}\otimes}$ in case $\underline{\mathbf{F}}$ is structured.

Lemma 10. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r \in \mathbb{N}$ and $0 \le \beta \le q$. Π^{batch^+} is perfectly correct for*

$$\Xi^{\mathsf{lin}}_{\hat{n},n,r,\beta} \to \Xi^{\mathsf{sum}\text{-}\otimes}_{\log_d m}\left[\Xi^{\mathsf{lin}}_{\underline{n},\underline{n},r,\beta}\right].$$

It is knowledge sound with its error $\kappa = (\hat{n} - \overline{n})/q^e$ for

$$\Xi^{\mathsf{sis}}_{\hat{n},n,r,2\beta'} \cup \Xi^{\mathsf{lin}}_{\hat{n},n,r,\beta'} \leftarrow \Xi^{\mathsf{sum}\text{-}\otimes}_{\log_d m}\left[\Xi^{\mathsf{lin}}_{\underline{n},\underline{n},r,\beta'}\right].$$

The communication cost is 0.

Proof. Correctness: First, it clearly holds that:

$$\left(\left(\overline{\mathbf{H}}, \overline{\mathbf{F}}, \overline{\mathbf{Y}}\right), \mathbf{W}\right) \in \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},r,\beta},$$

$$\sum_{\mathbf{z}\in[d]^{\log_d m}} h(\mathsf{LDE}[\mathbf{MW}])(\mathbf{z}) = \sum_{\mathbf{z}\in[d]^{\log_d m}} \mathsf{LDE}[\mathbf{c}^\mathsf{T}\underline{\mathbf{HF}}](\mathbf{z}) \cdot \mathsf{LDE}[\mathbf{W}](\mathbf{z}).$$

Let $\mathbf{h}^\mathsf{T} \coloneqq \mathbf{c}^\mathsf{T}\underline{\mathbf{HF}} \in \mathcal{R}_q^{1\times m}$ and $(\mathbf{w}_j^*)_{j\in[m]} \in \mathbf{W}^\mathsf{T}$.
Since

$$(\mathbf{h}^\mathsf{T} \cdot \mathbf{W})^\mathsf{T} = (\sum_{j\in[m]} \mathbf{h}_j^\mathsf{T} \cdot \mathbf{w}_j^{*\mathsf{T}})^\mathsf{T} = \sum_{j\in[m]} \mathbf{h}_j \cdot \mathbf{w}_j^* = \sum_{\mathbf{z}\in[d]^{\log_d m}} \mathsf{LDE}[\mathbf{h}](\mathbf{z}) \cdot \mathsf{LDE}[\mathbf{W}](\mathbf{z}),$$

where $j = \sum_{k\in[\log_d m]} \mathbf{z}_k d^k$, the following holds:

$$\mathbf{s} = (\mathbf{c}^\mathsf{T}\underline{\mathbf{HF}}\mathbf{W})^\mathsf{T} = (\mathbf{h}^\mathsf{T} \cdot \mathbf{W})^\mathsf{T} = \sum_{\mathbf{z}\in[d]^{\log_d m}} \mathsf{LDE}[(\mathbf{c}^\mathsf{T}\underline{\mathbf{HF}})^\mathsf{T}](\mathbf{z}) \cdot \mathsf{LDE}[\mathbf{W}](\mathbf{z}).$$

Knowledge Soundness:

Without loss of generality, suppose $\mathcal{P}^*$ is deterministic and succeeds with a probability ϵ. $\mathcal{E}$ runs $\mathcal{P}^*$ on a random challenge $\mathbf{c}$. If the prover fails, $\mathcal{E}$ aborts. On success, by the definition of the relation $\Xi^{\mathsf{sum}\text{-}\otimes}$, $\mathcal{E}$ obtains $(\mathbf{s}, (\overline{\mathbf{H}}, \overline{\mathbf{F}}, \overline{\mathbf{Y}}, \mathbf{I}_m), \mathbf{W}) \in \Xi^{\mathsf{sum}\text{-}\otimes}_{\log_d \underline{n}} \left[\Xi^{\mathsf{lin}}_{\overline{n},\overline{n},r,\beta'}\right]$ satisfying:

$$((\overline{\mathbf{H}}, \overline{\mathbf{F}}, \overline{\mathbf{Y}}), \mathbf{W}) \in \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},r,\beta'}, \quad \mathbf{s} = \sum_{\mathbf{z}\in[d]^{\log_d m}} \mathsf{LDE}[(\mathbf{c}^{\mathsf{T}}\underline{\mathbf{HF}})^{\mathsf{T}}](\mathbf{z}) \cdot \mathsf{LDE}[\mathbf{W}](\mathbf{z}).$$

If $((\mathbf{H}, \mathbf{F}, \mathbf{Y}), \mathbf{W})$ satisfies $\Xi^{\mathsf{lin}}_{\hat{n},n,r,\beta'}$, the extractor outputs $\mathbf{W}$ and terminates. Otherwise, it re-runs $\mathcal{P}^*$ on fresh challenges $\mathbf{c}'$ until it obtains the second accepting transcript $(\mathbf{s}', (\overline{\mathbf{H}}, \overline{\mathbf{F}}, \overline{\mathbf{Y}}, \mathbf{I}_m), \mathbf{W}') \in \Xi^{\mathsf{sum}\text{-}\otimes}_{\log_d \underline{n}} \left[\Xi^{\mathsf{lin}}_{\overline{n},\overline{n},r,\beta'}\right]$ satisfying:

$$((\overline{\mathbf{H}}, \overline{\mathbf{F}}, \overline{\mathbf{Y}}), \mathbf{W}') \in \Xi^{\mathsf{lin}}_{\overline{n},\overline{n},r,\beta}, \quad \mathbf{s}' = \sum_{\mathbf{z}\in[d]^{\log_d m}} \mathsf{LDE}[(\mathbf{c}'^{\mathsf{T}}\underline{\mathbf{HF}})^{\mathsf{T}}](\mathbf{z}) \cdot \mathsf{LDE}[\mathbf{W}'](\mathbf{z}).$$

From both transcripts, we obtain $\overline{\mathbf{HF}}\mathbf{W} = \overline{\mathbf{HF}}\mathbf{W}' \leftrightarrow \overline{\mathbf{HF}}(\mathbf{W}' - \mathbf{W}) = \mathbf{0}$. The extractor checks if $\mathbf{W}' \neq \mathbf{W}$. If so, let $\mathbf{w}$ be any non-zero column of $\mathbf{W}' - \mathbf{W}$, then $\overline{\mathbf{F}}\mathbf{w} = \mathbf{0}$ so $\mathcal{E}$ extracts $(\overline{\mathbf{F}}, \mathbf{w}) \in \Xi^{\mathsf{sis}}_{2\beta'}$ and terminates.

Consider the case $\mathbf{W}' = \mathbf{W}$ and $\underline{\mathbf{HF}}\mathbf{W} \neq \underline{\mathbf{Y}}$. From the correctness analysis and the conditions of the extracted witnesses,

$$(\mathbf{s}'^{\mathsf{T}} =) \quad \mathbf{c}'^{\mathsf{T}}\underline{\mathbf{Y}} = \mathbf{c}'^{\mathsf{T}}\underline{\mathbf{HF}}\mathbf{W},$$

but this, due to the Schwartz-Zippel lemma and since $\mathbf{c}'$ is independent of the first transcript, happens only with a probability of at most $\kappa := {}^{\hat{n} - \overline{n}}/_{q^e}$.

Expected runtime of $\mathcal{E}$. We argue that Π^{batch^+} admits an expected polynomial-time knowledge extractor when the prover $\mathcal{P}^*$ succeeds with probability ϵ. The argument follows the same reasoning as in Lemma 3 and requires no further elaboration. □

Remark 4. Although h function in Π^{batch^+} contains $\mathsf{LDE}[(\mathbf{c}^{\mathsf{T}}\underline{\mathbf{HF}})^{\mathsf{T}}]$, which is not row-tensor, the verifier can succinctly evaluate it at any point $\mathbf{r} \in \mathcal{R}_q^{\log_d m}$ at the end of the sumcheck protocol Π^{sum}. The efficiency follows from the fact that $\mathsf{LDE}[(\mathbf{c}^{\mathsf{T}}\underline{\mathbf{HF}})^{\mathsf{T}}]$ can be viewed as a linear combination of $\mathsf{LDE}[\underline{\mathbf{f}}_i^{\mathsf{T}}]$ for $i \in [\underline{n}]$, i.e. of low-degree extensions of the rows of $\underline{\mathbf{F}}$. More precisely, the time to compute $\mathsf{LDE}[(\mathbf{c}^{\mathsf{T}}\underline{\mathbf{HF}})^{\mathsf{T}}]$ is $\mathcal{O}(\underline{n} \log_d m)$ since each row $\underline{\mathbf{f}}_j^{\mathsf{T}}$ in $\underline{\mathbf{F}}$ has a tensor structure, and it takes time $\mathcal{O}(\log_d m)$ to compute an evaluation of a low-degree extension taken over the row-tensor by the mixed-product property.

We combine Π^{batch^+} with $\Pi^{\mathsf{lde}\text{-}\otimes}$ and Π^{sum} to obtain a batching protocol Π^{batch^*} from Ξ^{lin} to Ξ^{lin}.

Lemma 11. *Let $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}, m, r \in \mathbb{N}$ and $0 \leq \beta \leq q$. Let $\mu = \log_d m$ We can define an alternative batching RoK $\Pi^{\mathsf{batch}^*} := \Pi^{\mathsf{batch}^+} \circ \Pi^{\mathsf{sum}} \circ \Pi^{\mathsf{lde}\text{-}\otimes}$. Π^{batch^*} is perfectly correct for*

$$\Xi^{\mathsf{lin}}_{\hat{n},n,r,\beta} \to \Xi^{\mathsf{lin}}_{\overline{n}+1,\overline{n}+1,r,\beta}$$

and knowledge sound with its error $\frac{\hat{n}-\overline{n}+q\varphi/e-1+\mu 2(d-1)}{q^e}$ for

$$\left(\Xi^{\mathsf{sis}}_{\hat{n},n,r,2\beta'} \cup \Xi^{\mathsf{lin}}_{\hat{n},n,r,\beta',t}\right) \leftarrow \Xi^{\mathsf{lin}}_{\overline{n}+1,\overline{n}+1,r,\beta'}$$

The communication cost is $d\mu e \log_q + 1 \log |\mathcal{R}_q|$.

Proof. The correctness and knowledge-soundness follow from the correctness of each composed RoKs described in Lemmas 2, 7 and 10. The individual degree of h is $2(d-1)$ as h takes the $(d-1)$-degree LDE of the witness and multiplies it by LDE of another vector. □

E Implementation details

In this section, we provide additional details regarding our implementation and the choices made to optimise performance. We also discuss the specific parameters selected for our experiments.

Parameters selection and composition strategy. Concrete numbers have been obtained via a script build on top of the script provided by [KLNO24, KLNO25]. This script serves as a tool to explore the parameter space and identify optimal configurations for the protocol, with respect to the communication complexity. The script, written in SageMath, counts the various costs of the protocol, with a particular focus on the communication complexity and the statistical/computational errors. In our estimates, we always assume that $\lambda = 128$ and $\kappa = 2^{-80}$. The script uses the Lattice Estimator [APS15] to estimate the security level of the parameters. For each experiment, we assume that the witness is sampled uniformly and bound the range of the coefficients of the witness by 2^{10}

For argument systems, while the theoretical composition strategy outlined in Section 5.2 is designed to minimise communication asymptotically, we slightly adapt it in our implementation to improve concrete efficiency (by moving $\Pi^{b\text{-decomp}}$ before Π^{norm}) in the same way as in [KLNO24]. In folding schemes, we exactly follow the composition strategy from Section 6, which coincides with the one optimised for concrete efficiency.

The script employed for the parameter selection is available at

`github.com/lattice-arguments/salsaa/blob/main/est/est.ipynb`.

Parallelisation and vectorised arithmetic. For the most optimal performance, we choose to work with power-of-two cyclotomic rings $\mathcal{R} = \mathbb{Z}[\zeta_{\mathfrak{f}}]$ where $\mathfrak{f} = 2^k$ for some integer k. Power-of-two cyclotomic rings admit efficient arithmetic using the Number-Theoretic Transform (NTT), which is particularly well-suited for vectorisation. This allows us to leverage SIMD instructions and hardware acceleration, leading to significant performance improvements. Our implementation utilises Intel's HEXL library, which provides AVX-512-accelerated operations for efficient polynomial arithmetic over these rings. We select modulus $q \approx 2^{50}$ so that the operations' runtime is further improved using AVX-512-IFMA instructions available on modern Intel processors. We note that the prover implementation is heavily parallelised and thus benefits from a large number of cores, reflecting common practical scenarios such as server-client architectures in verifiable computation. The parallelisation of our algorithms has been optimized with careful attention to minimal memory overhead and to the cache locality of individual threads. Further performance gains could be achieved by enforcing even stricter locality, for instance through custom parallelised routines that incorporate NUMA-aware CPU pinning. The verifier, by contrast, requires minimal memory and remains efficient even on systems with a moderate or low number of cores, making it well-suited for client settings where devices are resource-constrained.

Cyclotomic rings and challenge spaces. Towards our performance disadvantage, we need to select a modulus q such that $\mathcal{R}_q \cong (\mathbb{F}_{q^2})^{\varphi/2}$, i.e. our cyclotomic ring splits into factors of degree 2. This is because:

- The knowledge error of the RoKs we use (e.g. $\Pi^{\otimes\mathsf{RP}}$, Π^{batch}) is at least $1/q^e$, where e is the residue degree of the ring modulo q. If $e = 1$, as in the case of fully splitting power-of-two cyclotomic rings, the knowledge error becomes too large.
- For Π^{fold} we require the existence of a low-norm strong sampling set [LS18, ALS20, BC25b] so that the inverse of any two elements is invertible. We require that the set is large enough to ensure a sufficiently small knowledge error and the operator norm of the challenged is small. Such set cannot exist if the ring fully splits as the largest subfield of such ring is $\mathbb{Z}_q$ itself, which itself is not large enough to provide a sufficiently small knowledge error. In other work, e.g. [BS23, NS24], the issue is avoided by using "almost-not-splitting" cyclotomic rings, i.e. rings that split into factors of large degree. Then, as shown in [LS18], low-norm challenges' differences are always invertible. However, such rings do not support NTT-based multiplication directly, leading to a significant computational overhead[10]. We take a different approach and sample challenges so that the coefficients are sampled uniformly from the ternary set $\{-1, 0, 1\}$. The set of challnges with ternary coefficients is not strong samplings sets per definition, but the probability of sampling elements so that the inverse of two is non-invertible is small. More precisely, the probability of sampling a two elements so that their difference is non-invertible is bounded in Lemma 32 of [BL25] by

$$\epsilon = \max_{j \in [k]} \left(\frac{1}{q} + \frac{1}{q} \sum_{t=1}^{q-1} \prod_{i=0}^{k-1} \left| p + (1-p) \cos\left(\frac{2\pi t r_j^i}{q} \right) \right| \right)^e,$$

[10] To circumvent this issue, in [BS23, NS24] the arithmetic is "lifted" to a higher modulus and then projected back to the original ring.

where p is the "bias" of the ternary distribution (i.e. the probability of sampling 0, in our case $p = 1/3$), r_j are the roots of $X^\varphi + 1$ in $\mathbb{Z}_q$, $k = \varphi/e$ is the number of irreducible factors of $X^\varphi + 1$ in $\mathbb{Z}_q[X]$, and e is the residue degree. Those result served as a direct generalisation of Lemma 3.2 of [ALS20]. Following the heuristic from Table 1 of [ALS20], we expect that the $\epsilon \approx \varphi/eq^e$. Therefore, while selecting $q \approx 2^{50}$ such that $q = \varphi + 1 \bmod \mathfrak{f}$ so that $e = 2$ and setting $p = 1/3$, we expect that the probability of sampling a non-invertible element is around 2^{-94}, which is negligible concerning the statistical error $\kappa = 2^{-80}$ [11]

Code organisation and toolset. Our implementation is written in Rust with the exception of C++ bindings to the HEXL library. The code is organised into several subroutines, each of which implements a specific component of the protocol. The protocol has been written in the interactive setting, and the non-interactive version could be obtained via Fiat-Shamir transformation using e.g. SHA-3 as the hash function. In the code we made a couple of changes compared to the theoretical description of the protocol, which are mostly implementation details and optimisations: (i) The norm has been tracked with respect to both ℓ_∞ and ℓ_2-norms (and the better guarantee is used for the parameter selection). (ii) The unstructured loop of the argument system has been omitted, since the witness is practically of negligible size (compared to the proof) and the runtime of the prover forms a geometric series through the rounds of the protocol (so the execution would be anyway extremely efficient). (iii) The sumcheck protocol over the extension field $\mathbb{F}_{q^e}$ has been executed over coefficients in $\mathbb{Z}_q$ (i.e. without lifting to the extension field) and then amplified for soundness by repeating the protocol multiple times with independent challenges. (iv) The Π^{batch} protocol has been omitted, since it does not impact the runtime. The system specification used for test execution is summarised in Table 4.

The code is made publicly available at `github.com/lattice-arguments/salsaa`. The repository includes scripts to obtain parameters, the experiments' code, and the reports generated during the execution.

Node type	Dell PowerEdge XE8640
Architecture	saphr avx2 h100 hopper 2024
CPU	2x48 core Xeon Platinum 8468 2.1GHz
Virtual cores (total/used)	192/96
Memory	1024GB DDR5-4800
OS	CentOS 7

Table 4: Specifications of the node used for experiments.

[11] We support this heuristic with empirical evidence:

`github.com/lattice-arguments/salsaa/blob/main/invertibility.ipynb`.