Orca And Dolphin: Efficient Bivariate And Multilinear Polynomial Commitment Schemes Under Standard Assumptions

March 17, 2026*

Helger Lipmaa [1]

University of Tartu, Tartu, Estonia

Abstract. Most polynomial commitment schemes have either superlinear prover time or superconstant argument size. Recently, Ganesh, Patranabis, and Singh introduced SamaritanPCS, and Eagen and Gabizon proposed Mercury. Both build on efficient univariate polynomial IOPs that lift univariate polynomial commitment schemes (PCSs) to the multilinear setting, enabling sum-check-based multilinear polynomial IOPs for prover-efficient zk-SNARKs with small communication. Since multilinear PCSs are fundamental building blocks of zk-SNARKs, they must be secure under minimal assumptions while remaining maximally efficient. However, SamaritanPCS and Mercury achieve knowledge soundness only in the joint random-oracle and algebraic-group model. We introduce Orca and Dolphin, optimized bivariate and multilinear PCSs, respectively. We prove that their interactive evaluation protocols have computational special soundness in the standard model, assuming that KZG satisfies binding and interpolation binding (both secure under ARSDH). Thus, they have knowledge soundness in the random oracle model. Both schemes can have a more efficient evaluation protocol that is knowledge sound in the joint random-oracle and algebraic-group model. Dolphin's evaluation phase is more efficient than either SamaritanPCS's or Mercury's.

Keywords: KZG · polynomial commitment scheme · polynomial IOP · special soundness · zk-SNARK

1 Introduction

Zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs, [Mic94,DL08,Gro10,Gro16]) have revolutionized the verification of complex computations by enabling a prover to convince a verifier of a computation's correctness using extremely succinct arguments. A common strategy in modern zk-SNARK design is to combine polynomial interactive oracle proofs (polynomial IOPs, [BFS20,CHM+20]) and polynomial commitment schemes (PCSs, [KZG10]). To minimize communication, one typically uses the univariate KZG polynomial commitment scheme [KZG10], which has constant communication

* Most of the work was done in 2025.

(the sum of the commitment and evaluation argument lengths). However, when using KZG, one has to use univariate polynomial IOPs, resulting in suboptimal prover performance, since the prover has to execute FFTs.

Multilinear polynomial IOPs are more prover-friendly than univariate ones due to the use of sum-check [LFKN90] and no need to interpolate the inputs, resulting in a linear-time prover. However, multilinear polynomial commitment schemes are traditionally less efficient. Up to recently, even the most verifier-efficient multilinear polynomial commitment schemes (for example, PST [PST13], Gemini [BCHO22], and Zeromorph [KT23]) required either a superlinear-time prover or $\Omega(\log n)$ communication. In 2025, Ganesh, Patranabis, and Singh (SamaritanPCS, [GPS25]) and Eagen and Gabizon (Mercury, [EG25]) introduced efficient univariate polynomial IOPs that enable the construction of a multilinear polynomial commitment scheme from a univariate one. SamaritanPCS and Mercury are the first known multilinear polynomial commitment schemes with linear prover computation and constant communication. Although they share core ideas, they use different optimization strategies. Moreover, SamaritanPCS is presented more modularly, with their multilinear polynomial commitment scheme being built on an efficient *bivariate* polynomial commitment scheme of Zapico et al. [ZBK+22, Section 4.6], where the latter is based on a univariate polynomial commitment scheme.

Subsequently, Singh, Patranabis, and Sinha [SPS25] used SamaritanPCS (or Mercury) to construct a sub-logarithmic-size, linear-time zk-SNARK from compressed sum-check. We refer to [GPS25,EG25] for more information on SamaritanPCS and Mercury, and also for an overview of the related work.

Unfortunately, both SamaritanPCS and Mercury have several issues.

Issue 1: Missing Details. Unfortunately, [ZBK+22] and the follow-up works [CGG+24,GPS25] do not fully specify their bivariate PCSs nor prove their security. Similarly, Mercury [EG25] does not fully spell out the concrete evaluation protocol and efficiency accounting needed for our comparison (we were only able to fill in the efficiency numbers in Table 2 due to [Gab]). While [GPS25] gives a full specification of SamaritanPCS, it only states its security claim but omits the proof. Such a situation is obviously unsatisfactory: to deploy basic primitives like polynomial commitment schemes, one must have both complete specifications *and* detailed, verifiable, security proofs. The unfortunate fact is that *none* of the mentioned papers provides all the details.

Issue 2: Security. SamaritanPCS and Mercury claim knowledge soundness in the joint random oracle and algebraic group model (AGM, [FKL18]). However, [GPS25] does not even include a knowledge-soundness proof. AGM[1] suffers from instantiability problems [Zha22], some of which can be mitigated via a proper interpretation of AGM [JM24]. Lipmaa, Parisella, and Siim (TCC 2023) [LPS23] show that AGM does not capture adversaries who can obliviously sample group elements without knowing their discrete logarithms. Since

[1] Many issues extend to the generic group model. For concreteness, we discuss AGM.

one can implement oblivious sampling efficiently in elliptic-curve groups using hash-and-increment or admissible encodings [Ica09,LPS23], this demonstrates the limitations of the AGM's modeling power. Similarly, oblivious sampling issues are known for the generic group model since [Bro01]. Lipmaa, Parisella, and Siim [LPS23] introduced AGM with oblivious sampling (AGMOS) to model adversaries with the power of oblivious sampling (see also [LPS25]).

Notably, it is often assumed that one can extract a polynomial from a KZG commitment C. While valid in the AGM, such an extraction fails when oblivious sampling is allowed. Lipmaa, Parisella, and Siim [LPS23] observed that extraction from C is possible only after C has been opened. They showed that the standard linearization trick (LinTrick [GWC19,CHM+20]), while secure in AGM, can fail in AGMOS. LinTrick underlies all highly optimized KZG-based zk-SNARKs like Plonk [GWC19], Marlin [CHM+20], Basilisk [RZ21], Lunar [CFF+21], and Vampire [LSZ22]. Subsequent cryptanalysis by Faonio, Fiore, and Russo [FFR24] and by Lipmaa, Parisella, and Siim (Crypto 2025, [LPS25]) established that LinTrick lacks computational special soundness [CDS94] in the standard model, unless computing the discrete logarithm is easy. Consequently, the flaw identified in AGM also compromises some real-world protocols, demonstrating that AGM analysis without oblivious sampling can be insufficient to prove security. This raises the question of whether KZG-based zk-SNARKs like Plonk [GWC19] or Marlin [CHM+20] possess knowledge soundness in the AGMOS or computational special soundness in the standard model (plus the random oracle model).

At Eurocrypt 2024, Lipmaa, Parisella, and Siim [LPS24] demonstrated how to overcome this problem "in theory" by proving that KZG has computational special soundness under a new but standard-looking falsifiable assumption, ARSDH. They then demonstrated how to modify KZG-based zk-SNARKs to achieve computational special soundness in the standard model, assuming only that KZG has computational special soundness. (Essentially, for this, one has to add a random opening of any polynomial commitment.)

In Crypto 2025, Lipmaa, Parisella, and Siim [LPS25] proved that the Plonk zk-SNARK [GWC19] has computational special soundness in the random oracle model under ARSDH and the additional falsifiable SplitRSDH assumption. They introduced a *sanitization technique* that adds a random batch opening to any unopened polynomial commitments. They proved that Plonk with sanitization (which adds one field element to the communication) has computational special soundness under ARSDH in the random-oracle model. This result relies on Plonk's fortunate design choices, specifically, that all relevant polynomial commitments are opened at a random position, as required by [LPS24]. It is not a priori clear that this result generalizes to other KZG-based zk-SNARKs or to interactive protocols in general.

A different aspect of security is tightness. [GPS25,EG25] claim (and [EG25] proves) the knowledge soundness of their interactive evaluation protocol. While one can obtain a non-interactive knowledge sound argument from an interactive one (in the random oracle model) after using the Fiat-Shamir transforma-

Table 1. Comparison of bivariate polynomial commitment schemes. $\mathcal{P}_{\mathsf{Eval}}$ and $\mathcal{V}_{\mathsf{Eval}}$ are the prover and the verifier of the evaluation phase. $|\mathsf{Eval.tr}|$ denotes the length of the prover's communication in the evaluation phase (the number of bytes is for the BLS12-381 curve [BGM17]). $\mathbb{F}$ is the scalar field. MSM = $\mathbb{G}_1$ multiscalar multiplication, KS = knowledge soundness, CSS = computational special soundness. $\mathsf{L}(\mathbb{A})$ stands for one element of $\mathbb{A}$. The communication does not include the verifier's messages.

	Security	$\lvert\mathsf{Eval.tr}\rvert$ (bytes)	$\mathcal{P}_{\mathsf{Eval}}$ computation	$\mathcal{V}_{\mathsf{Eval}}$ computation
Orca_{agm}	KS (AGM)	$3 \cdot \mathsf{L}(\mathbb{G}_1)$ (144)	$2\times$ n-wide MSM, $\leq 2\times$ n_y-wide MSM	4-wide MSM, 2-wide multipairing
Orca	CSS	$3 \cdot \mathsf{L}(\mathbb{G}_1) + 1 \cdot \mathsf{L}(\mathbb{F})$ (176)	$2\times$ n-wide MSM, $\leq 2\times$ n_y-wide MSM	4-wide MSM, 2-wide multipairing

tion [FS87], the resulting non-interactive argument can have non-tight security. (See [AFK22] for discussion.)

Issue 3: Efficiency. Zk-SNARKs are widely implemented. The choice of which zk-SNARK to implement depends heavily on the concrete security of the zk-SNARK: concrete factors like a 20% efficiency difference playing a crucial role in adoption. Both SamaritanPCS and Mercury are quite efficient (see Table 2), but their efficiency is not close to that of KZG for the verifier. For example, the length of the evaluation argument is 8 group elements and 6 field elements in Mercury and 7 group elements and 1 field element in SamaritanPCS. Moreover, although they share the same underlying theoretical basis, they employ very different, partially orthogonal optimization tricks. Finally, both are optimized for security in the AGM. For example, since both SamaritanPCS and Mercury rely strongly on LinTrick without opening all polynomial commitments, they might require sanitization. If so, their no-AGM variants might be even less efficient.

Main Open Question. Can SamaritanPCS and Mercury be modified to exhibit tight knowledge soundness in the random oracle model under a falsifiable assumption (no AGM or knowledge assumptions) without incurring significant efficiency losses? Even more ambitiously, can one design a new polynomial commitment scheme that offers both stronger security and greater efficiency?

Our contributions. We introduce two polynomial commitment schemes: a bivariate scheme (Orca, building on [ZBK+22]) and a multilinear scheme (Dolphin, building on [GPS25,EG25]). Our first contribution is to give *fully specified* variants Orca_{agm} and Dolphin_{agm} of both Orca and Dolphin whose evaluation phases have knowledge soundness in the joint random oracle model and the AGM. Differing from [GPS25,EG25], we fully specify both the AGM and standard-model variants and provide a complete security analysis of the standard-model variants.

Orca_{agm} is very efficient (see Table 1); since preceding schemes like [ZBK+22] are not fully specified, so a direct comparison to them is impossible. Dolphin_{agm} is significantly more efficient than SamaritanPCS and Mercury.

Our second contribution is a route to computational special soundness via the sanitization technique of [LPS25]. We prove computational special soundness instead of just knowledge soundness since it results in tight knowledge soundness after Fiat-Shamir [AFK22]. By using sanitization, the prover can batch-open (all) previously unopened commitments. In the no-AGM variants, sanitization increases communication by exactly one field element. Importantly, sanitization leaves computation essentially unchanged up to low-order terms. We prove that after sanitization, the evaluation protocol Eval of both Orca and Dolphin has computational special soundness in the standard model assuming only that KZG is binding and interpolation binding [AJM+23] (see Section 2). Following Attema, Fehr, and Klooß [AFK22], Eval has *tight* knowledge soundness in the random oracle model. Thus, Orca and Dolphin have strong security (extractability) in the random-oracle model, assuming KZG has weak security (binding and interpolation binding). Having polynomial commitment schemes with strong security can enable the design of more efficient zk-SNARKs.

Putting these pieces together yields a bivariate PCS, Orca, that is slightly less efficient than Orca_{agm}, and a multilinear PCS, Dolphin, that improves on the best known multilinear PCSs while offering stronger security. See Tables 1 and 2. Recall that for BLS12-381 we have $\mathsf{L}(\mathbb{F}) = 32$ and $\mathsf{L}(\mathbb{G}_1) = 48$ bytes [BGM17]. Since [ZBK+22,CGG+24,GPS25] do not fully specify their bivariate PCSs (and do not prove security), we cannot compare against those schemes with complete certainty. Providing a detailed analysis is part of our contribution.

1.1 Our Techniques

On Orca. Orca is inspired by earlier bivariate polynomial commitment schemes. The latter have independent applications, but Orca also serves as a pedagogical example of our techniques. Zapico et al. [ZBK+22] (see also Choudhuri et al. [CGG+24]) proposed a bivariate commitment scheme where to commit to $Q(X, Y)$ with $\deg_X Q \leq \mathsf{n}_\mathsf{x} - 1$ and $\deg_Y Q \leq \mathsf{n}_\mathsf{y} - 1$, the prover commits to $\hat{f}(Y) := Q(Y^{\mathsf{n}_\mathsf{y}}, Y)$. In the evaluation phase, when asked to open $Q(\mathsf{x}, \mathsf{y})$ to ϑ, the prover first commits to $\hat{p}(Y)$ as $[\hat{p}(\tau)]_1$, where τ is a trapdoor.[2] It then proves the correctness of $\hat{p}(Y)$ by computing $[\hat{r}(\tau)]_1 \leftarrow [(\hat{f}(\tau) - \hat{p}(\tau))/(\tau^{\mathsf{n}_\mathsf{y}} - \mathsf{x})]_1$ and the correctness of ϑ by computing $[\hat{\ell}(\tau)]_1 \leftarrow [(\hat{p}(\tau) - \vartheta)/(\tau - \mathsf{y})]_1$. The verifier checks both by using KZG-like verifications. Zapico et al. [ZBK+22, Section 4.6] mention that the second check has to be done with respect to the degree bound $\deg \hat{p}(Y) \leq \mathsf{n}_\mathsf{y} - 1$ but do not explain how to achieve it. Thus, it is difficult to estimate the efficiency of their scheme.

Moreover, since [ZBK+22] mentions the degree bound very briefly, it is easy to miss it. Indeed, Choudhuri et al. [CGG+24] use the scheme of [ZBK+22] but without mentioning that one needs to prove that $\hat{p}(Y)$ has a low degree. This

[2] We use the standard additive bracket notation, see Section 2.

version of the commitment scheme (without the degree check) is insecure. Indeed, in the AGM, the verifier's checks guarantee that $(Y^{\mathsf{n_y}} - \mathsf{x}) \mid (\hat{p}(Y) - \hat{f}(Y))$ and $\hat{p}(\mathsf{y}) = \vartheta$. Both checks are satisfied by $\hat{p}(Y) = Q(\mathsf{x}, Y) + (Y^{\mathsf{n_y}} - \mathsf{x})t(Y)$ for any $t(Y)$. If $t(\mathsf{y})$ is non-zero and $\mathsf{x} \neq \mathsf{y}^{\mathsf{n_y}}$, $\hat{p}(\mathsf{y}) = \vartheta$ guarantees that $Q(\mathsf{x}, \mathsf{y}) + (\mathsf{y}^{\mathsf{n_y}} - \mathsf{x})t(\mathsf{y}) = \vartheta$ but not that $Q(\mathsf{x}, \mathsf{y}) = \vartheta$.

Ganesh, Patranabis, and Singh [GPS25] generalize the scheme of [ZBK+22] to arbitrary underlying univariate polynomial schemes. As [ZBK+22], they mention that a degree test on $\hat{p}(Y)$ is needed, but do not explain how to do it. Notably, none of the papers mentioned fully specify or prove the security of the commitment scheme (even in the AGM).

Orca solves the above problems by fully specifying a degree test for the polynomial $\hat{p}(Y)$. For the sake of concrete efficiency, Orca's evaluation phase is different. In the evaluation phase, when opening $Q(\mathsf{x}, \mathsf{y})$ to ϑ, the prover commits to $\hat{r}(Y)$ and a shifted polynomial $\bar{\ell}(Y) := \Delta_{\mathsf{y}}(Y)\hat{\ell}(Y)$, where $\Delta_{\mathsf{y}}(Y)$ is a degree-$((\mathsf{n_x}\mathsf{n_y} - 1) - (\mathsf{n_y} - 2))$ polynomial satisfying $\gcd(\Delta_{\mathsf{y}}(Y), Y - \mathsf{y}) = 1$. This guarantees that $\deg(\hat{\ell}) \leq \mathsf{n_y} - 2$ and thus $\deg(\hat{p}) \leq \mathsf{n_y} - 1$. It does not commit to $\hat{p}(Y)$. Instead, the prover uses the linearization trick to prove the correctness of the opening; see Section 3. Without sanitization, we obtain a variant Orca_{agm} of Orca that is knowledge sound in the AGM. (Since our primary interest is in no-AGM Orca, we do not prove this fact, but it follows directly by observing the proof of the following claim.) With sanitization, we obtain Orca and prove that it has special soundness in the standard model, assuming only that KZG is binding and interpolation binding [AJM+23]. ([KLPS26] proved that interpolation binding does not imply binding.) See Section 2 for the definition of interpolation binding and Section 3 for details on Orca.

On Dolphin. As in Mercury and SamaritanPCS, the multilinear commitment of $\tilde{f}(\boldsymbol{X}) = \sum_{i=1}^{n} f_i \tilde{\mathsf{eq}}_i(\boldsymbol{X})$ (a multilinear extension of $\boldsymbol{f}$) is the univariate commitment to $\hat{f}(X) = \sum_{i=1}^{n} f_i X^{i-1}$ (the univariate extension of $\boldsymbol{f}$). Here, $\tilde{\mathsf{eq}}$ are the multilinear Lagrange polynomials (see Section 2). The commitment does not require the prover to compute FFTs. Since the prover directly commits to the coefficient vector, committing to zero elements is free (they can be skipped), and committing to small coefficients is cheaper than committing to large ones.

The evaluation protocol Eval of Dolphin is based on the same techniques as those of SamaritanPCS and Mercury, but is significantly optimized (and achieves computational special soundness in the standard model under falsifiable assumptions). We will give a thorough derivation of Dolphin from first principles (without assuming the knowledge of previous work) in 4. In the rest of the introduction, we will explain how Dolphin differs from SamaritanPCS and Mercury. This description assumes that the reader knows SamaritanPCS and Mercury.

In SamaritanPCS [GPS25], the prover batch-commits to a linear combination of certain polynomials and then opens that combination. To obtain computational special soundness, we must "unbatch" these commitments, which costs two additional polynomial commitments (two group elements), and forces the prover to open at least five commitments. We instead apply our optimized san-

Table 2. Comparison of linear-time prover and constant-size multilinear polynomial commitment schemes. See Table 1 for the notation.

	Security	$\lvert\mathsf{Eval.tr}\rvert$ (bytes)	$\mathcal{P}_{\mathsf{Eval}}$ computation	$\mathcal{V}_{\mathsf{Eval}}$ computation
Mercury [EG25]	KS (AGM)	$8 \cdot \mathsf{L}(\mathbb{G}_1) + 6 \cdot \mathsf{L}(\mathbb{F})$ (576)	n-wide MSM, $(n-\sqrt{n})$-wide MSM, $6\times$ $\sqrt{n}$-wide MSM	7-wide MSM, 2-wide MSM, 2-wide multipairing
SamaritanPCS [GPS25]	KS (AGM)	$7 \cdot \mathsf{L}(\mathbb{G}_1) + 1 \cdot \mathsf{L}(\mathbb{F})$ (368)	n-wide MSM, $(n-\sqrt{n})$-wide MSM, $5\times$ $\sqrt{n}$-wide MSM	8-wide MSM, 3-wide multipairing
Dolphin	CSS	$6 \cdot \mathsf{L}(\mathbb{G}_1) + 1 \cdot \mathsf{L}(\mathbb{F})$ (320)	n-wide MSM, $(n-\sqrt{n})$-wide MSM, $4\times$ $\sqrt{n}$-wide MSM	7-wide MSM, 2-wide multipairing
Dolphin$_{agm}$	KS (AGM)	$6 \cdot \mathsf{L}(\mathbb{G}_1)$ (288)	n-wide MSM, $(n-\sqrt{n})$-wide MSM, $4\times$ $\sqrt{n}$-wide MSM	7-wide MSM, 2-wide multipairing

itization technique: we batch-open all yet-unopened polynomials and perform linearization using the same challenge. This reduces the added communication to a single field element rather than five.

We add optimizations to offset the added overhead. Following Mercury, we assume the commitment key has the correct length (i.e., it bounds the degree of the committed univariate polynomial), avoiding a costly degree check and reducing the number of verifier pairings. We split the n-dimensional vector into $\mathsf{n}_\mathsf{x} = \sqrt{n}$-element subvectors, whereas SamaritanPCS allows arbitrary $\mathsf{n}_\mathsf{x} < n$. If we are satisfied with a knowledge-soundness proof in the AGM, Dolphin communicates one fewer field element than SamaritanPCS. We also include several optimizations in SamaritanPCS that are not common in Mercury. For example, we use an IPA-based opening of one of the polynomial commitments, which can be batched with the existing IPA in SamaritanPCS. This reduces the number of distinct evaluation points by 1 compared to Mercury. We use another SamaritanPCS's optimization to cut out another evaluation point.

Table 2 compares Mercury, SamaritanPCS, and Dolphin (both the AGM and no-AGM variants). Notably, Dolphin$_{agm}$ reduces Mercury's evaluation-phase communication by a factor of (exactly) two, while the verifier of both Dolphin and Dolphin$_{agm}$ is approximately 15% faster based on our back-of-the-envelope calculations. Extreme succinctness is crucial in many practical applications, as evidenced by the popularity of Groth16 [Gro16] in industry. In all cases, the commitment can be computed as an n-wide MSM. In addition, in all cases in Table 2 one directly commits to the coefficient vector. Thus, unlike KZG, committing to zero coefficients is free, and committing to a sparse polynomial (few non-zero coefficients) is considerably cheaper than committing to a dense one. Both properties are crucial in many applications [DP25,ST25].

We prove that interactive Orca and Dolphin have computational special soundness in the standard model, assuming that KZG has both binding and interpolation binding [AJM+23]. i.e., under the falsifiable ARSDH assumption [LPS24]. Curiously, evaluation binding is not needed. By Attema, Fehr, and Klooß [AFK22], it follows that, after applying Fiat-Shamir [FS87], both schemes are tightly knowledge-sound in the random oracle model under ARSDH. Our

arguments adapt techniques from [KLPS26][3], relying on KZG's binding and interpolation binding [AJM+23] properties.

Additional Contributions. Our paper has three more independent contributions. First, we extend the techniques of [KLPS26], which addressed only the result of a standard compilation of polynomial IOPs with polynomial commitment schemes, by proving computational special soundness for interactive protocols that exploit the homomorphism and (evaluation, interpolation, degree) binding properties of KZG. While Chiesa et al. [CGKY25] recently introduced a framework for state-restoration soundness using state-restoration function binding of polynomial commitment schemes, we show (following [KLPS26]) that establishing *computational special soundness* under more standard and easier to prove notions of binding and interpolation binding is straightforward. Interestingly, our proofs do not rely on evaluation binding. Second, we first optimize Orca and Dolphin for efficiency in the AGM. We use the sanitization technique of [LPS25] to obtain computational special soundness in the standard model with the cost of adding a single field element to the communication (and a small amount of field operations to the computation) of both Orca and Dolphin. Third, we optimize the sanitization technique itself by providing a rule that requires fewer polynomials to be opened.

Security Lesson. The research direction initiated by [LPS24] and developed by [LPS25,CGKY25,KLPS26] demonstrates that the polynomial commitment scheme does not have to have extractability for the zk-SNARK to have knowledge soundness. It allows one to avoid the AGM in KZG-based zk-SNARKs and can have applications elsewhere. In particular, there is no reason *not* to aim for computational special soundness or, possibly, a similar notion like state-restoration or round-by-round soundness.

2 Preliminaries

For field elements r_i, let $\mathsf{Van}(r_1, \ldots, r_k)$ be the Vandermonde matrix

$$\mathsf{Van}(r_1, \ldots, r_k) = \begin{pmatrix} 1 & r_1 & \ldots & r_1^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & r_k & \ldots & r_k^{k-1} \end{pmatrix} \in \mathbb{F}^{k \times k} \quad .$$

Van is invertible given r_i are pairwise distinct. Let λ denote the security parameter. By $f(\lambda) \approx_\lambda 0$, we mean that f is a negligible function. PPT (resp. DPT) stands for probabilistic (resp. deterministic) polynomial time. $\mathbb{F}$ is a finite field of prime order $|\mathbb{F}| = 2^{\Theta(\lambda)}$ while all polynomials have degree $\mathsf{poly}(\lambda)$.

[3] There, the authors established that the mentioned three binding properties of KZG, together with the polynomial IOP's special soundness, are sufficient to prove that the compiled interactive protocol has computational special soundness. However, [KLPS26] only consider the compilation of abstract polynomial IOPs with abstract polynomial commitment schemes.

Univariate Polynomials. $\mathbb{F}[X]$ is the polynomial ring in variable X over the field $\mathbb{F}$ and $\mathbb{F}_{\leq d}[X] \subset \mathbb{F}[X]$ is the set of polynomials of at most degree d. Given a tuple $T = (\delta_i, \upsilon_i)_{i=1}^n$, $\hat{f}(X) = \mathsf{Interp}((\delta_i, \upsilon_i)_{i=1}^n, X)$ is the interpolation polynomial of T, the unique degree-$\leq (n-1)$ polynomial satisfying $\hat{f}(\delta_i) = \upsilon_i$ for all $i \in [n]$.

Multilinear Polynomials. Let $\boldsymbol{X} = (X_1, \ldots, X_\mu)$. To make the dimension explicit, we sometimes write $\boldsymbol{X}^\mu = (X_1, \ldots, X_\mu)$. Denote by $\mathbb{F}_{\leq \boldsymbol{d}}[\boldsymbol{X}^\mu]$ the set of μ-variate polynomials $f(\boldsymbol{X}^\mu)$, where $\deg_{X_i}(f) \leq d_i$. A multivariate polynomial $f(\boldsymbol{X})$ is *multilinear* if it is linear in every variable. Let $\mathbb{B} = \{0,1\} \subset \mathbb{F}$. The *multilinear extension* of $f : \mathbb{B}^\mu \to \mathbb{F}$ is the unique multilinear polynomial $\tilde{f} : \mathbb{F}^\mu \to \mathbb{F}$ that agrees with f on $\mathbb{B}^\mu$. Let $\mathsf{eq} : \mathbb{B}^\mu \times \mathbb{B}^\mu \to \{0,1\}$ be the function satisfying $\mathsf{eq}(\boldsymbol{x}, \boldsymbol{y}) = 1$ iff $\boldsymbol{x} = \boldsymbol{y}$. The multilinear extension $\tilde{\mathsf{eq}} : \mathbb{F}^\mu \times \mathbb{F}^\mu \to \mathbb{F}$ (the multilinear Lagrange polynomial) of eq is $\tilde{\mathsf{eq}}(\boldsymbol{X}, \boldsymbol{Y}) := \prod_{i=1}^\mu (X_i Y_i + (1-X_i)(1-Y_i))$. It is well known that $\tilde{f}(\boldsymbol{X}) = \sum_{\boldsymbol{y} \in \mathbb{B}^\mu} f(\boldsymbol{y}) \tilde{\mathsf{eq}}(\boldsymbol{X}, \boldsymbol{y})$.

Bilinear Groups. A bilinear group generator $\mathsf{PGen}(1^\lambda)$ returns $\mathsf{pp} = (\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, [1]_2)$, where $\mathbb{F}$ is a scalar field of prime order $|\mathbb{F}|$, all three of $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are additive[4] cyclic (thus, abelian) groups of order $|\mathbb{F}|$, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear pairing, and $[1]_\iota$ is a fixed generator of $\mathbb{G}_\iota$ for $\iota \in \{1, 2, T\}$. While $[1]_\iota$ is part of pp, we often give input to algorithms explicitly for clarity. The bilinear pairing is of Type-3, that is, there is no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. We use the common bracket notation: for $\iota \in \{1, 2, T\}$ and $a \in \mathbb{F}$, we write $[a]_\iota$ to denote $a[1]_\iota$. Denote $e([a]_1, [b]_2)$ by $[a]_1 \bullet [b]_2$ and assume $[1]_T = [1]_1 \bullet [1]_2$. Thus, $[a]_1 \bullet [b]_2 = [ab]_T$ for any $a, b \in \mathbb{F}$. We assume that multiplication has higher priority than pairing, thus $a[b]_1 \bullet [c]_2 = (a[b]_1) \bullet [c]_2$. Let $[\boldsymbol{b}]_\iota = [b_1, \ldots, b_n]_\iota$ for $\boldsymbol{b} \in \mathbb{F}^n$.

We recall the following falsifiable assumptions PDL and ARSDH.

Definition 1 (PDL). PGen *is* (d_1, d_2)-PDL (Power Discrete Logarithm, [Lip12]) secure *if for any PPT* $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{pdl}}_{d_1,d_2,\mathsf{PGen},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\mathcal{A}(\mathsf{pp}, [1, \tau, \ldots, \tau^{d_1}]_1, [1, \tau, \ldots, \tau^{d_2}]_2) = \tau \mid \mathsf{pp} \leftarrow \mathsf{PGen}(1^\lambda); \tau \leftarrow_\$ \mathbb{F}\right] \approx_\lambda 0 \ .$$

Definition 2. PGen *is* d-ARSDH (Adaptive Rational Strong Diffie-Hellman, [LPS24]) *secure in* $\mathbb{G}_1$ *if for any PPT* $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{arsdh}}_{\mathsf{PGen},d,\mathbb{G}_1,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} \mathcal{S} \subset \mathbb{F} \wedge |\mathcal{S}| = d+1 \wedge [g]_1 \neq [0]_1 \wedge \\ [g]_1 \bullet [1]_2 = [\varphi]_1 \bullet [\prod_{s \in \mathcal{S}} (\tau - s)]_2 \end{array} \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{PGen}(1^\lambda); \tau \leftarrow_\$ \mathbb{F}; \\ \mathsf{ck} \leftarrow ([1, \tau, \ldots, \tau^d]_1, [1, \tau]_2); \\ (\mathcal{S}, [g, \varphi]_1) \leftarrow \mathcal{A}(\mathsf{ck}) \end{array} \right] \approx_\lambda 0 \ .$$

2.1 Polynomial Commitment Schemes

A polynomial commitment scheme enables a prover to succinctly open evaluations of a committed polynomial. We present such schemes with a non-interactive

[4] In the literature, $\mathbb{G}_T$ is often denoted as a multiplicative group. However, we find the use of uniformly additive notation more convenient.

commitment phase and an interactive evaluation phase. We extend the definition from [NS24] to multivariate polynomials; see also [BFS20].

A polynomial commitment scheme (with non-interactive commitment and interactive evaluation) over $\mathbb{F}$ is a tuple $\mathsf{PC} = (\mathsf{KGen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Eval})$ of PPT algorithms, where:

- $(\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KGen}(1^\lambda, \mu, \boldsymbol{d})$**:** Inputs a security parameter λ, number of variables μ and upper bounds $d_i \in \mathbb{N}$, $i \in [\mu]$, on the degree of each variable X_i for a μ-variate polynomial, KGen generates a commitment key (that also contains public parameters like the group description) and a trapdoor. In the case of univariate polynomial commitment schemes, we omit the parameter μ. We will implicitly assume ck contains parameters pp.
- $(C, \mathsf{dc}) \leftarrow \mathsf{Com}(\mathsf{ck}, f)$**:** Inputs a commitment key ck, and a polynomial $f \in \mathbb{F}_{\leq \boldsymbol{d}}[\boldsymbol{X}^\mu]$, Com outputs a commitment C to the polynomial f, and additionally a decommitment state dc.
- $b \leftarrow \mathsf{Open}(\mathsf{ck}, C, f, \mathsf{dc})$**:** Inputs a commitment key ck, the commitment C and the decommitment state dc, a polynomial $f \in \mathbb{F}_{\leq \boldsymbol{d}}[\boldsymbol{X}^\mu]$, Open outputs a bit indicating accept or reject.
- **Evaluation:** a public coin interactive protocol $(\mathsf{tr}, b) \leftarrow \langle \mathcal{P}_{\mathsf{Eval}}(\mathbb{w}) \leftrightarrow \mathcal{V}_{\mathsf{Eval}} \rangle (\mathsf{ck}, \mathbb{x})$ between a PPT prover $\mathcal{P}_{\mathsf{Eval}}$ and a PPT verifier $\mathcal{V}_{\mathsf{Eval}}$, where $\mathbb{x} = (C, \boldsymbol{\delta}, v_f)$ and $\mathbb{w} = (f, \mathsf{dc})$. The parties share a common input, including a commitment key ck, commitment C, evaluation point $\boldsymbol{\delta}$, and claimed evaluation v_f. $\mathcal{P}_{\mathsf{Eval}}$ also inputs the opening $f \in \mathbb{F}_{\leq \boldsymbol{d}}[\boldsymbol{X}^\mu]$ of C and a decommitment state dc. At the end of the protocol, $\mathcal{V}_{\mathsf{Eval}}$ outputs a transcript tr (all messages sent between the prover and the verifier) and a bit b, with $b = 1$ indicating accepting that $f(\boldsymbol{\delta}) = v_f$, or $b = 0$ indicating rejecting it.

A polynomial commitment scheme with an interactive opening phase can satisfy the following security properties (completeness, computational special soundness, and knowledge soundness) for any fixed μ and $\boldsymbol{d}$. After that, we will define a separate collection of security requirements for schemes with a non-interactive opening phase.

Definition 3 (Completeness). *For all $f(\boldsymbol{X}) \in \mathbb{F}_{\leq \boldsymbol{d}}[\boldsymbol{X}^\mu]$, for all $\boldsymbol{\delta} \in \mathbb{F}^\mu$,*

$$\Pr\left[\begin{array}{c|c} \mathsf{Open}(\mathsf{ck}, C, f, \mathsf{dc}) = 1 \\ \wedge\, b = 1 \end{array} \middle| \begin{array}{c} (\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KGen}(1^\lambda, \mu, \boldsymbol{d}); \\ (C, \mathsf{dc}) \leftarrow \mathsf{Com}(\mathsf{ck}, f); \\ v_f \leftarrow f(\boldsymbol{\delta}); \mathbb{x} \leftarrow (C, \boldsymbol{\delta}, v_f); \mathbb{w} \leftarrow (f, \mathsf{dc}); \\ (\mathsf{tr}, b) \leftarrow \langle \mathcal{P}_{\mathsf{Eval}}(\mathbb{w}) \leftrightarrow \mathcal{V}_{\mathsf{Eval}} \rangle (\mathsf{ck}, \mathbb{x}) \end{array}\right] = 1 \ .$$

Let $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_{\mathsf{rnds}}) \in \mathbb{N}^{\mathsf{rnds}}$. A *$\boldsymbol{\kappa}$-tree of transcripts* for a $(2 \cdot \mathsf{rnds}+1)$-move public-coin interactive argument $\Pi = (\mathcal{P}, \mathcal{V})$ is a set of $K = \prod_{i=1}^{\mathsf{rnds}} \kappa_i$ transcripts arranged in a tree structure. The tree nodes correspond to the prover's messages (where the first message includes the common input, and some prover's messages can be empty strings) and the edges to the verifier's challenges. Every node at depth i has precisely κ_i children corresponding to κ_i *pairwise distinct* challenges. Every transcript corresponds to exactly one path from the root node to a leaf node. We refer to the corresponding tree of challenges as a *$\boldsymbol{\kappa}$-tree of challenges.*

We will also refer to a $\boldsymbol{\kappa}$-tree of transcripts or a $\boldsymbol{\kappa}$-tree of challenges.

Definition 4 ($\boldsymbol{\kappa}$-out-of-$\boldsymbol{N}$ Special Soundness). *Let $\boldsymbol{\kappa}, \boldsymbol{N} \in \mathbb{N}^{\mathsf{rnds}}$. A $(2 \cdot \mathsf{rnds} + 1)$-move public-coin interactive argument $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathcal{R}$, where $\mathcal{V}$ samples the i-th challenge from a set of cardinality $N_i \geq \kappa_i$ for $1 \leq i \leq \mathsf{rnds}$, has $\boldsymbol{\kappa}$-out-of-$\boldsymbol{N}$ special soundness [CDS94] if there exists a polynomial time extractor $\mathcal{E}_{\mathsf{css}}$ that, on input a statement $\mathbb{x}$ and a $\boldsymbol{\kappa}$-tree of accepting transcripts outputs a witness $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$. We also say Π has $\boldsymbol{\kappa}$-special soundness.*

We assume $\boldsymbol{N}$ is implicit. Π has $\boldsymbol{\kappa}$-*computational special soundness ($\boldsymbol{\kappa}$-CSS)* if it has $\boldsymbol{\kappa}$-special soundness, given that a PPT adversary constructed the tree of accepting transcripts. We denote by $\mathsf{Adv}^{\mathrm{ss}}_{\mathsf{PGen},\Pi,\boldsymbol{\kappa},\mathcal{E}_{\mathsf{css}},\mathcal{A}_{\mathsf{css}}}(\lambda)$ the probability that $\mathcal{E}_{\mathsf{css}}$ does not succeed, given an input Tree generated by $\mathcal{A}_{\mathsf{css}}$.

In the current paper, we prove computational special soundness of evaluation protocols Eval in the standard model. According to Attema, Fehr, and Klooß [AFK22], Eval will have knowledge soundness in the random-oracle model. Such polynomial commitment schemes are usually called *extractable*.

Binding properties for non-interactive PCSs. In a *non-interactive univariate polynomial commitment scheme* like the KZG, the evaluation protocol has only one message, with $(\pi, b) \leftarrow \mathsf{Eval}(\mathsf{ck}, C, \mathbb{x}; \mathbb{w})$, where π is a bitstring (an evaluation argument) and $b = 1$ iff the verifier accepts. Let $\mathsf{Eval}(\mathsf{ck}, C, \delta, \hat{f}(X))$ be the non-interactive algorithm that creates $\upsilon_f \leftarrow \hat{f}(\delta)$ and π and $\mathsf{Vf}(\mathsf{ck}, C, \delta, \upsilon_f, \pi)$ the non-interactive algorithm that checks if π is the correct argument. From such commitment schemes, we need the following security requirements. Note that we only consider deterministic polynomial commitment schemes.

Definition 5 (Binding). *A univariate non-interactive polynomial commitment scheme is* binding *if for all λ, d, and PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{b}}_{\mathsf{PGen},\mathsf{PC},d,\mathcal{A}}(\lambda) :=$*

$$\Pr\left[\begin{array}{c} f_0 \neq f_1 \ \wedge \ f_0, f_1 \in \mathbb{F}_{\leq d}[X] \wedge \\ (C, \mathsf{dc}_0) = \mathsf{Com}(\mathsf{ck}, f_0) \wedge \\ (C, \mathsf{dc}_1) = \mathsf{Com}(\mathsf{ck}, f_1) \end{array} \middle| \begin{array}{c} (\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KGen}(1^\lambda, d); \\ (C, \mathsf{dc}_0, \mathsf{dc}_1, f_0, f_1) \leftarrow \mathcal{A}(\mathsf{ck}) \end{array}\right] \approx_\lambda 0 \ .$$

In our applications, $\mathsf{dc}_i = f_i$, so $\mathcal{A}$ will not become weaker when required to output dc_i.

Abraham et al. [AJM+23] (Crypto 2023) say that a non-interactive polynomial commitment scheme is *interpolation binding* if, given enough accepting evaluations concerning the same polynomial commitment, the interpolated polynomials obtained from these evaluations must agree with the commitment.

Definition 6 (Interpolation binding [AJM+23]). *A univariate non-interactive polynomial commitment scheme is* interpolation binding *if for all λ, d, and PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{ib}}_{\mathsf{PGen},\mathsf{PC},d,\mathcal{A}}(\lambda) :=$*

$$\Pr\left[\begin{array}{c} (\forall i \neq j)(\delta_i \neq \delta_j) \wedge \\ (\forall i \in [d+1])\mathsf{Vf}(\mathsf{ck}, C, \delta_i, \upsilon_{f,i}, \pi_i) = 1 \\ \wedge (C, \mathsf{dc}) \neq \mathsf{Com}(\mathsf{ck}, \hat{f}) \end{array} \middle| \begin{array}{c} (\mathsf{ck}, \mathsf{tk}) \leftarrow \mathsf{KGen}(1^\lambda, d); \\ (C, \mathsf{dc}, (\delta_i, \upsilon_{f,i}, \pi_i)_{i=1}^{d+1}) \leftarrow \mathcal{A}(\mathsf{ck}); \\ \hat{f} \leftarrow \mathsf{Interp}((\delta_i, \upsilon_{f,i})_{i=1}^{d+1}, X) \end{array}\right]$$

is negligible.

KZG *Commitment Scheme.* The KZG [KZG10] non-interactive univariate polynomial commitment scheme is defined as follows:

$\mathsf{KZG.KGen}(1^\lambda, d)$**:** $\mathsf{pp} \leftarrow \mathsf{PGen}(1^\lambda)$; $\mathsf{tk} = \tau \leftarrow\!\!\$\, \mathbb{F}$; $\mathsf{ck} \leftarrow (\mathsf{pp}, [1, \tau, \ldots, \tau^d]_1, [1, \tau]_2)$; return $(\mathsf{ck}, \mathsf{tk})$.

$\mathsf{KZG.Com}(\mathsf{ck}, \hat{f})$**:** return $C \leftarrow [\hat{f}(\tau)]_1 = \sum_{j=0}^{d} \hat{f}_j [\tau^j]_1$. (Since $\mathsf{dc} = \hat{f}$, we simplify the notation by not returning dc.)

$\mathsf{KZG.Eval}(\mathsf{ck}, [\hat{f}(\tau)]_1, \delta, \hat{f})$**:** $\upsilon_f \leftarrow \hat{f}(\delta)$; $\hat{\Pi}(X) \leftarrow \frac{\hat{f}(X) - \upsilon_f}{X - \delta}$; return $(\upsilon_f, [\hat{\Pi}(\tau)]_1)$.

$\mathsf{KZG.Vf}(\mathsf{ck}, [\hat{f}(\tau)]_1, \delta, \upsilon_f, [\hat{\Pi}(\tau)]_1)$**:** return 1 iff $([\hat{f}(\tau)]_1 - \upsilon_f [1]_1) \bullet [1]_2 = [\hat{\Pi}(\tau)]_1 \bullet ([\tau]_2 - \delta[1]_2)$. (Can be optimized to $([\hat{f}(\tau)]_1 - \upsilon_f [1]_1 + \delta[\hat{\Pi}(\tau)]_1) \bullet [1]_2 = [\hat{\Pi}(\tau)]_1 \bullet [\tau]_2$.)

KZG is binding under d-PDL and interpolation binding under d-ARSDH [CGKY25]. Since d-PDL follows from d-ARSDH [LPS24], KZG is binding and interpolation binding under d-ARSDH.

A transcript of a non-interactive polynomial commitment scheme Π is $\mathsf{tr} = (C, \delta, \upsilon_f, \pi)$. The transcript tr is accepting if $\mathsf{Vf}(\mathsf{ck}, \mathsf{tr} = (C, \delta, \upsilon_f, \pi)) = 1$. Π is *transcript-homomorphic* [LPS25], if: if $(C_j, \delta, \upsilon_{f,j}, \pi_j)$ is an accepting transcript for the same ck and δ for every j, then so is $(\sum s_j C_j, \delta, \sum s_j \upsilon_{f,j}, \sum s_j \pi_j)$ for any $s_j \in \mathbb{F}$ for the same ck. KZG is transcript-homomorphic since $\hat{f} \mapsto [f(\tau)]_1$, $\hat{f} \mapsto f(\delta)$, and $\hat{f} \mapsto [(\hat{f}(\tau) - \hat{f}(\delta))/(\tau - \delta)]_1$ (for fixed δ) are linear maps.

3 Orca: Bivariate Polynomial Commitment Scheme

In their construction of a multilinear polynomial commitment scheme with constant communication, Ganesh, Patranabis, and Singh [GPS25] use a bivariate polynomial commitment scheme with constant communication. Based on Zapico et al. [ZBK+22] (see also [CGG+24]), they proposed one based on KZG. Ganesh, Patranabis, and Singh do not provide all details of their scheme. Motivated by [ZBK+22], we propose a new bivariate polynomial commitment scheme, Orca. We first optimize it for the AGM security and then modify it by using the sanitization technique [LPS25] so that it will have computational special soundness [CDS94] assuming that KZG is binding and interpolation binding. On the other hand, [CGG+24,GPS25] claimed (but did not prove) that their schemes are extractable in the AGM.

For the reader's convenience, we mostly adopt the notation of [GPS25]. The prover commits to a bivariate polynomial $Q(X, Y)$, with $\deg_X(Q) \le \mathsf{n_x} - 1$ and $\deg_Y(Q) \le \mathsf{n_y} - 1$, by KZG-committing to the univariate polynomial $\hat{f}(Y) := Q(Y^{\mathsf{n_y}}, Y)$. In the evaluation protocol, to open $Q(X, Y)$ at (x, y), the prover computes the polynomial $\hat{p}(Y) = Q(\mathsf{x}, Y) \in \mathbb{F}_{\le \mathsf{n_y}-1}[Y]$ together with a KZG argument polynomial $\hat{r}(Y)$ that $\hat{p}(Y)$ is correct, and then a KZG argument polynomial $\hat{\ell}(Y)$ ascertaining that $\hat{p}(\mathsf{y}) = \vartheta$. It then proves that

i $\hat{f}(Y) - ((Y^{\mathsf{n_y}} - \mathsf{x})\hat{r}(Y) + (Y - \mathsf{y})\hat{\ell}(Y) + \vartheta)$ is a zero polynomial, and
ii $\hat{p}(Y)$ has degree $\le \mathsf{n_y} - 1$.

Note that i does not explicitly mention $\hat{p}$. We refer to [GPS25] for more intuition.

Degree test. We prove $\deg(\hat{p}) \leq \mathsf{n_y} - 1$ by committing to $\bar{\ell}(Y) := \Delta_{\mathsf{y}}(Y)\hat{\ell}(Y)$ instead of $\hat{\ell}(Y)$, where $\Delta_{\mathsf{y}}(Y)$ is a degree-$(d - (\mathsf{n_y} - 2))$ polynomial satisfying $\gcd(\Delta_{\mathsf{y}}(Y), Y - \mathsf{y}) = 1$. We set $\Delta_{\mathsf{y}}(Y) := Y^{d-(\mathsf{n_y}-2)} - 1$, if $\mathsf{y} = 0$, and $\Delta_{\mathsf{y}}(Y) := Y^{d-(\mathsf{n_y}-2)}$, if $\mathsf{y} \neq 0$. We need $\gcd(\Delta_{\mathsf{y}}(Y), Y - \mathsf{y}) = 1$ so that from $\Delta_{\mathsf{y}}(Y)$ dividing $(Y - \mathsf{y})\bar{\ell}(Y)$ we can conclude $\bar{\ell}(Y) = \Delta_{\mathsf{y}}(Y)\hat{\ell}(Y)$ for some polynomial $\hat{\ell}(Y)$; see the proof of Theorem 1. Since Item i is checked by the verifier at a random point δ, for this to work, we need to sample δ such that $\Delta_{\mathsf{y}}(\delta) \neq 0$; this excludes at most $d - (\mathsf{n_y} - 2) \ll |\mathbb{F}|$ elements from the sampling space $|\mathbb{F}|$.

On LinTrick: Proving Computational Special Soundness. The well-known linearization trick (LinTrick, [CHM+20,GWC19]) allows one to prove that $\hat{d}(X) \equiv 0$ by linearizing $\hat{d}$. For example, if $\hat{d}(X) = a(X)b(X) - c(X)$, then the prover opens $a(X)$ and the linearized polynomial $\upsilon_a b(X) - c(X)$ at δ to υ_a and 0, respectively. The verifier checks both evaluation arguments. In Orca, the verifier can open public polynomials at $Y = \delta$ without the prover's assistance. (For example, $Y^{\mathsf{n_y}} - \mathsf{x}$ is opened to $\delta^{\mathsf{n_y}} - \mathsf{x}$.) After that, the prover opens the linearized polynomial $\hat{f}(Y) - ((\delta^{\mathsf{n_y}} - \mathsf{x})\hat{r}(Y) + (\delta - \mathsf{y})\hat{\ell}(Y) + \vartheta)$ at $Y = \delta$ to 0.

As proven in [LPS25], under the discrete logarithm assumption, LinTrick cannot have computational special soundness in the standard model. To still obtain it, we modify the bivariate polynomial commitment scheme. A naïve approach is to open the commitments $[\bar{\ell}(\tau)]_1$, $[\hat{f}(\tau)]_1$ and $[\hat{r}(\tau)]_1$ individually, adding three field elements to the communication. Following the sanitization technique of [LPS25], the prover instead opens a random linear combination of all yet-unopened polynomials by using new sanitizer challenges σ and ϱ. We optimize this technique by observing that the pre-batch linearization polynomial is a linear batch of affine functionals, computed from committed polynomials. Thus, it is not needed to open the linearization polynomial and *all* committed polynomials. Instead, one can leave unopened one committed polynomial in each batch (in total, one in Orca and two in Dolphin). In security proofs (Theorems 1 and 2), we virtually open the remaining polynomials using all other openings.

This results in the linearization polynomial

$$\begin{aligned}\hat{L}(Y) &= (\hat{f}(Y) - ((\delta^{\mathsf{n_y}} - \mathsf{x})\hat{r}(Y) + (\delta - \mathsf{y})\bar{\ell}(Y)/\Delta_{\mathsf{y}}(\delta)) - \vartheta) \\ &\quad + \varrho \cdot (\bar{\ell}(Y) + \sigma\hat{r}(Y) - \upsilon_\sigma) \\ &= \hat{f}(Y) + (\varrho\sigma - (\delta^{\mathsf{n_y}} - \mathsf{x}))\hat{r}(Y) + (\varrho - (\delta - \mathsf{y})/\Delta_{\mathsf{y}}(\delta))\bar{\ell}(Y) - (\vartheta + \varrho\upsilon_\sigma)\end{aligned} \tag{1}$$

that the prover opens at a random point δ to 0. Note that the sanitization batch does not contain $\hat{f}(Y)$. The AGM variant remains unchanged (no sanitization) while the no-AGM variant contains the blue terms. (Note that we use orange and blue to distinguish AGM and no-AGM parts.)

Construction. Fig. 1 depicts two interactive bivariate polynomial commitment schemes, one that has knowledge soundness in the AGM (Orca_{agm}) and another one that has computational special soundness in the standard model (Orca). One also needs the random oracle model when using Fiat-Shamir.

$\mathsf{KGen}(1^\lambda, 2, (\mathsf{n_x}-1, \mathsf{n_y}-1))$:
return $(\mathsf{ck}, \mathsf{tk}) \leftarrow \mathrm{KZG}.\mathsf{KGen}(1^\lambda, d = n-1)$ for $n = \mathsf{n_x}\mathsf{n_y}$.

$\mathsf{Com}(\mathsf{ck}, Q(X,Y))$: // $\deg_X(Q) \le \mathsf{n_x}-1$ and $\deg_Y(Q) \le \mathsf{n_y}-1$
$\hat{f}(Y) \leftarrow Q(Y^{\mathsf{n_y}}, Y) \in \mathbb{F}_{\le n-1}[Y]$. // $\hat{f}(Y) = \sum_{i=0}^{d} f_i Y^i$
Return $[\hat{f}(\tau)]_1 \leftarrow \mathrm{KZG}.\mathsf{Com}(\mathsf{ck}, \hat{f}) = \sum_{i=1}^{n} f_i [\tau^{i-1}]_1$.

$\langle \mathcal{P}_{\mathsf{Eval}}(Q(X,Y)) \leftrightarrow \mathcal{V}_{\mathsf{Eval}} \rangle([\hat{f}(\tau)]_1, (\mathsf{x}, \mathsf{y}), \vartheta)$: // Prove $Q(\mathsf{x},\mathsf{y}) = \vartheta$

1. $\mathcal{P}_{\mathsf{Eval}}$: $\hat{p}(Y) \leftarrow Q(\mathsf{x}, Y) \in \mathbb{F}_{\le \mathsf{n_y}-1}[Y]$.
   $R(X,Y) \leftarrow (Q(X,Y) - \hat{p}(Y))/(X-\mathsf{x})$. // Synthetic division in $(\mathbb{F}[Y])[X]$
   $\hat{r}(Y) \leftarrow R(Y^{\mathsf{n_y}}, Y) \in \mathbb{F}_{\le n-\mathsf{n_y}-1}[Y]$.
   $\hat{\ell}(Y) \leftarrow (\hat{p}(Y) - \vartheta)/(Y-\mathsf{y}) \in \mathbb{F}_{\le \mathsf{n_y}-2}[Y]$.
   $\bar{\ell}(Y) \leftarrow \Delta_{\mathsf{y}}(Y)\hat{\ell}(Y) \in \mathbb{F}_{\le d}[Y]$.
   Send $[\hat{r}(\tau), \bar{\ell}(\tau)]_1$. // $[\hat{r}(\tau)]_1$ — **expensive computation**
2. $\mathcal{V}_{\mathsf{Eval}}$: send $\delta \leftarrow_{\$} \mathbb{F} \setminus \Delta_{\mathsf{y}}^{-1}(0)$.

---

3. $\mathcal{V}_{\mathsf{Eval}}$: $\sigma \leftarrow_{\$} \mathbb{F}$.

---

4. $\mathcal{P}_{\mathsf{Eval}}$: send $\upsilon_\sigma \leftarrow \bar{\ell}(\delta) + \sigma\hat{r}(\delta)$.
5. $\mathcal{V}_{\mathsf{Eval}}$: send $\varrho \leftarrow_{\$} \mathbb{F}$. // Sent after δ, σ for CSS/Fiat-Shamir security

---

6. $\mathcal{P}_{\mathsf{Eval}}$: Compute $\hat{L}(Y)$ as in Eq. (1).
   $\hat{H}(Y) \leftarrow \hat{L}(Y)/(Y-\delta)$.
   Send $[\hat{H}(\tau)]_1$. // $[\hat{H}(\tau)]_1$ — **expensive computation**
7. $\mathcal{V}_{\mathsf{Eval}}$: check // $\mathrm{KZG}.\mathsf{Open}(\mathsf{ck}, [\hat{L}(\tau)]_1, \delta, 0, [\hat{H}(\tau)]_1)$

$$\begin{pmatrix} [\hat{f}(\tau)]_1 + (\varrho\sigma - (\delta^{\mathsf{n_y}} - \mathsf{x}))[\hat{r}(\tau)]_1 \\ +(\varrho - (\delta - \mathsf{y})/\Delta_{\mathsf{y}}(\delta))[\bar{\ell}(\tau)]_1 \\ -(\vartheta + \varrho\upsilon_\sigma)[1]_1 + \delta[\hat{H}(\tau)]_1 \end{pmatrix} \bullet [1]_2 = [\hat{H}(\tau)]_1 \bullet [\tau]_2 \ .$$

Fig. 1. A bivariate polynomial commitment scheme: A version Orca_{agm} with knowledge soundness in the AGM and Orca with computational special soundness in the standard model. In the no-AGM variant, we add sanitization (blue highlighting).

Evaluation Phase's Efficiency. In Orca_{agm}, the evaluation transcript is $([\hat{r}(\tau), \bar{\ell}(\tau)]_1; \delta; [\hat{H}(\tau)]_1)$. The prover sends three group elements. The prover executes one $(n - \mathsf{n_y})$-wide MSM, one $(\mathsf{y} \neq 0)$ or two $(\mathsf{y} = 0)$ $\mathsf{n_y}$-wide MSMs, and one n-wide MSM to compute $[\hat{r}(\tau), \bar{\ell}(\tau), \hat{H}(\tau)]_1$, respectively. ($[\bar{\ell}(\tau)]_1$ can be computed by an $\le 2\mathsf{n_y}$-wide MSM since $\bar{\ell}(Y)$ has $\le 2\mathsf{n_y}$ non-zero coefficients.) The verifier executes one four-wide MSM and a two-wide multipairing.

In Orca (the variant, CSS in the standard model), the transcript

$$([\hat{r}(\tau), \bar{\ell}(\tau)]_1; \delta; \sigma; \upsilon_\sigma; \varrho; [\hat{H}(\tau)]_1)$$

contains one more prover's field element υ_σ. (Crucially, the verifier sends σ a round after δ; thus, after Fiat-Shamir, $\sigma = \mathsf{H}(\ldots, \delta)$.) The computational complexity of the prover and the verifier stays essentially unchanged (the difference is in a small number of field operations).

Since $[\hat{f}(\tau)]_1$ is not multiplied by a scalar, the verifier executes a 4-wide MSM and a two-wide multipairing. The interactive protocol has three messages in the AGM version and six in the CSS version. See Table 1 for an efficiency summary.

3.1 Computational Special Soundness of Orca

Theorem 1. *Let $d = n - 1 \ll |\mathbb{F}|$ and $n = \mathsf{n_x}\mathsf{n_y}$. Assume $n/|\mathbb{F}|$ is negligible.*[5] *Assume KZG is binding and interpolation binding; both hold under the (d-)ARSDH assumption. Let $\boldsymbol{\kappa} = (\kappa_\delta = 2d+3, \kappa_\sigma = 2, \kappa_\varrho = 2)$, and $\boldsymbol{N} = (N_\delta = |\mathbb{F} \setminus \Delta_{\mathsf{y}}^{-1}(0)|, N_\sigma = |\mathbb{F}|, N_\varrho = |\mathbb{F}|)$. The interactive CSS-version of Orca has $\boldsymbol{\kappa}$-out-of-$\boldsymbol{N}$ computational special soundness in the standard model.*

Intuitively, interpolation binding is needed since Orca's evaluation phase is a commit-and-prove SNARK, with the commitment being the statement. Thus, the extractor needs to extract a polynomial that agrees with the statement. Note that evaluation binding is not needed.

Before proving Theorem 1, we state and prove the following lemma that is used in both Theorems 1 and 2.

Lemma 1. *Let $d = \mathsf{n_x}\mathsf{n_y} - 1$. Let $\hat{f}^*(Y) = \sum_{i=0}^{\mathsf{n_y}-1} f_i(Y^{\mathsf{n_y}})Y^i \in \mathbb{F}_{\leq d}[Y]$ with $\deg(f_i(X)) \leq \mathsf{n_x}-1$. Let $Q(X,Y) := \sum_{i=0}^{\mathsf{n_y}-1} f_i(X)Y^i$. If for some $\hat{r}^*(Y) \in \mathbb{F}_{\leq d}[Y]$ and $\mathsf{x} \in \mathbb{F}$, $\hat{f}^*(Y) = (Y^{\mathsf{n_y}} - \mathsf{x})\hat{r}^*(Y) + \hat{p}^*(Y)$, where $\hat{p}^*(Y) \in \mathbb{F}_{\leq \mathsf{n_y}-1}[Y]$, then $Q(Y^{\mathsf{n_y}}, Y) = \hat{f}^*(Y)$ and $Q(\mathsf{x}, Y) = \hat{p}^*(Y)$.*

Proof. Note that $\deg_X(Q) < \mathsf{n_x}$ and $\deg_Y(Q) < \mathsf{n_y}$. By construction, $Q(Y^{\mathsf{n_y}}, Y) = \hat{f}^*(Y)$. Performing Euclidean division of Q by $X - \mathsf{x}$ in $(\mathbb{F}[Y])[X]$, we get that there exist unique $R(X,Y) \in (\mathbb{F}[Y])[X]$ and $P(Y) \in \mathbb{F}[Y]$ satisfying $Q(X,Y) = (X - \mathsf{x})R(X,Y) + P(Y)$. Evaluating at $X = Y^{\mathsf{n_y}}$ gives

$$\hat{f}^*(Y) = (Y^{\mathsf{n_y}} - \mathsf{x})R(Y^{\mathsf{n_y}}, Y) + P(Y) \ .$$

The lemma assumption together with the uniqueness of quotient and remainder yields $R(Y^{\mathsf{n_y}}, Y) = \hat{r}^*(Y)$ and $P(Y) = \hat{p}^*(Y)$. Evaluating at $X = \mathsf{x}$, $Q(\mathsf{x}, Y) = P(Y) = \hat{p}^*(Y)$. □

Proof (Of Theorem 1). Let $\boldsymbol{k}_j = (k_1, \ldots, k_j)$ for every j. In the $\boldsymbol{\kappa}$-computational special-soundness proof of Orca, we construct KZG binding/interpolation-binding adversaries $\mathcal{B}_{\mathsf{b}}$, $\mathcal{B}_{\mathsf{ib}}$, and an extractor $\mathcal{E}_{\mathsf{css}}$. The CSS adversary outputs an accepting $\boldsymbol{\kappa}$-tree $\mathsf{Tree} = (\mathsf{tr}_{\boldsymbol{k}_3})_{\boldsymbol{k}_3 \in [\kappa_\delta] \times [\kappa_\sigma] \times [\kappa_\varrho]}$, where

$$\mathsf{tr}_{\boldsymbol{k}_3} = ([\hat{f}(\tau)]_1, \mathsf{x}, \mathsf{y}; \vartheta, [\hat{r}(\tau), \bar{\ell}(\tau)]_1, \delta_{k_1}; \sigma_{\boldsymbol{k}_2}; \upsilon_{\sigma,\boldsymbol{k}_2}; \varrho_{\boldsymbol{k}_3}; [\hat{H}_{\boldsymbol{k}_3}(\tau)]_1) \ .$$

Thus, σ is in the next tree layer after δ. Denote

$$[g_{1,k_1}(\tau)]_1 := [\hat{f}(\tau) - \big((\delta_{k_1}^{\mathsf{n_y}} - \mathsf{x})\hat{r}(\tau) + (\delta_{k_1} - \mathsf{y})\bar{\ell}(\tau)/\Delta_{\mathsf{y}}(\delta_{k_1}) + \vartheta\big)]_1 \ ,$$

[5] More precisely, for Tree to exist, we need that $\kappa_\delta < N_\delta$, which implies $|\mathbb{F}| > 3d - \mathsf{n_y} + 5$. For the knowledge error to be negligible, we need $n/|\mathbb{F}|$ to be negligible.

$$[g_{2,\boldsymbol{k}_2}(\tau)]_1 := [\bar{\ell}(\tau)+\sigma_{\boldsymbol{k}_2}\hat{r}(\tau)]_1 \ .$$

(Since $\mathcal{V}_{\mathsf{Eval}}$ samples δ with $\Delta_{\mathsf{y}}(\delta) \neq 0$, divisions by $\Delta_{\mathsf{y}}(\delta_{k_1})$ are well-defined.)

Fix any $\boldsymbol{k}_2 \in [\kappa_\delta] \times [\kappa_\sigma]$. By the verifier's acceptance (see Fig. 1), we have

$$([g_{1,k_1}(\tau)]_1 + \varrho_{\boldsymbol{k}_3}([g_{2,\boldsymbol{k}_2}(\tau) - \upsilon_{\sigma,\boldsymbol{k}_2}]_1)) \bullet [1]_2 = [\hat{H}_{\boldsymbol{k}_3}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ .$$

Let $\boldsymbol{A}_{\boldsymbol{k}_2} = \mathsf{Van}(\varrho_{\boldsymbol{k}_2 1}, \varrho_{\boldsymbol{k}_2 2}) \in \mathbb{F}^{2\times 2}$ (this is why $\kappa_\varrho = 2$); by transcript distinctness, $\det \boldsymbol{A}_{\boldsymbol{k}_2} \neq 0$. Define

$$[\hat{\boldsymbol{H}}^*_{\boldsymbol{k}_2}(\tau)]_1 := \boldsymbol{A}^{-1}_{\boldsymbol{k}_2} \begin{bmatrix} \hat{H}_{\boldsymbol{k}_2 1}(\tau) \\ \hat{H}_{\boldsymbol{k}_2 2}(\tau) \end{bmatrix}_1 .$$

By KZG's transcript-homomorphism,

$$[g_{1,k_1}(\tau)]_1 \bullet [1]_2 = [\hat{H}^*_{1,\boldsymbol{k}_2}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ , \tag{2}$$

$$[g_{2,\boldsymbol{k}_2}(\tau) - \upsilon_{\sigma,\boldsymbol{k}_2}]_1 \bullet [1]_2 = [\hat{H}^*_{2,\boldsymbol{k}_2}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ . \tag{3}$$

Here, it is important that υ_σ does not depend on ϱ (is chosen earlier).

Fix any $k_1 \in [\kappa_\delta]$ and $k_2 \in [\kappa_\sigma]$. By the definition of $[g_{2,\boldsymbol{k}_2}(\tau)]_1$ and Eq. (3),

$$[\bar{\ell}(\tau) + \sigma_{\boldsymbol{k}_2}\hat{r}(\tau) - \upsilon_{\sigma,\boldsymbol{k}_2}]_1 \bullet [1]_2 = [\hat{H}^*_{2,\boldsymbol{k}_2}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ .$$

Next, fix any k_1. Let $\boldsymbol{B}_{k_1} = \mathsf{Van}(\sigma_{k_1 1}, \sigma_{k_1 2}) \in \mathbb{F}^{2\times 2}$ (this is why $\kappa_\sigma = 2$); by transcript distinctness, $\det \boldsymbol{B}_{k_1} \neq 0$. Define

$$\begin{bmatrix} \bar{\ell}_{k_1} \\ r_{k_1} \end{bmatrix}_1 := \boldsymbol{B}^{-1}_{k_1} \begin{bmatrix} \upsilon_{\sigma,k_1 1} \\ \upsilon_{\sigma,k_1 2} \end{bmatrix}_1, \quad [\hat{\boldsymbol{H}}^{**}_{k_1}(\tau)]_1 := \boldsymbol{B}^{-1}_{k_1} \begin{bmatrix} \hat{H}^*_{2,(k_1,1)}(\tau) \\ \hat{H}^*_{2,(k_1,2)}(\tau) \end{bmatrix}_1 .$$

By KZG's transcript-homomorphism,

$$[\bar{\ell}(\tau) - \bar{\ell}_{k_1}]_1 \bullet [1]_2 = [\hat{H}^{**}_{k_1 1}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ , \tag{4}$$

$$[\hat{r}(\tau) - r_{k_1}]_1 \bullet [1]_2 = [\hat{H}^{**}_{k_1 2}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ . \tag{5}$$

$\mathcal{B}_{\mathsf{b}}$, $\mathcal{B}_{\mathsf{ib}}$, and $\mathcal{E}_{\mathsf{css}}$ compute the following interpolating polynomials for an arbitrary subset $\mathcal{T} \subset [\kappa_\delta]$ of size $d+1$:

1. $\bar{\ell}^*(Y) \in \mathbb{F}_{\leq d}[Y]$ interpolates $(\delta_{k_1}, \bar{\ell}_{k_1})_{k_1 \in \mathcal{T}}$ (see Eq. (4)),
2. $\hat{r}^*(Y) \in \mathbb{F}_{\leq d}[Y]$ interpolates $(\delta_{k_1}, r_{k_1})_{k_1 \in \mathcal{T}}$ (see Eq. (5)),
3. $\hat{f}^*(Y) \in \mathbb{F}_{\leq d}[Y]$ interpolates $(\delta_{k_1}, f(\delta_{k_1}) := (\delta^{\mathsf{n_y}}_{k_1} - \mathsf{x})r_{k_1} + \frac{\delta_{k_1} - \mathsf{y}}{\Delta_{\mathsf{y}}(\delta_{k_1})} \cdot \bar{\ell}_{k_1} + \vartheta)_{k_1 \in \mathcal{T}}$ (see Eq. (2)).

Since KZG is transcript-homomorphic, we can construct KZG evaluation arguments to show that $\hat{t}^*(\delta_{k_1})$ is correct for each $\hat{t}^* \in \{\bar{\ell}^*, \hat{r}^*, \hat{f}^*\}$. For example,

$$[\hat{f}^*(\tau) - f(\delta_{k_1})]_1 \bullet [1]_2 = [(\delta^{\mathsf{n_y}}_{k_1} - \mathsf{x})\hat{H}^{**}_{k_1 2}(\tau) + \tfrac{\delta_{k_1} - \mathsf{y}}{\Delta_{\mathsf{y}}(\delta_{k_1})} \cdot \hat{H}^{**}_{k_1 1}(\tau)]_1 \bullet [\tau - \delta_{k_1}]_2 \ .$$

Up to now, everything worked at the "polynomial IOP" level and did not involve polynomial commitments, except that we assumed KZG is transcript-homomorphic to be able to compute such KZG evaluation arguments. Next, we use the constructed evaluation arguments to rule out attack vectors that involve breaking the KZG.

- If $[\hat{f}(\tau)]_1 \neq [\hat{f}^*(\tau)]_1$, $[\bar{\ell}(\tau)]_1 \neq [\bar{\ell}^*(\tau)]_1$, or $[\hat{r}(\tau)]_1 \neq [\hat{r}^*(\tau)]_1$, we set $\mathsf{bad}_{\mathsf{ib}} = \mathsf{true}$. If $\mathsf{bad}_{\mathsf{ib}} = \mathsf{true}$, then $\mathcal{B}_{\mathsf{ib}}$ can break KZG's interpolation binding. $\mathcal{E}_{\mathsf{css}}$ aborts when $\mathsf{bad}_{\mathsf{ib}} = \mathsf{true}$.
- Let $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}}$ be the case that some $\hat{t}^*$, where $\hat{t}^* \in \{\hat{r}^*, \bar{\ell}^*, \hat{f}^*\}$, does not interpolate the corresponding data (e.g., $(\delta_{k_1}, \bar{\ell}_{k_1})$) on the whole interval $[\kappa_\delta]$. Then, there exists a subset of $[\kappa_\delta]$ of size $d+1$, such that the interpolation polynomial $\hat{t}'$ through those points is not equal to $\hat{t}^*$. If $[\hat{t}(\tau)]_1 \neq [\hat{t}'(\tau)]_1$, then we have broken interpolation binding. Otherwise, we can break binding by returning $([\hat{t}(\tau)]_1, \hat{t}^*(X), \hat{t}'(X))$.
  If $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{true}$, then either $\mathcal{B}_{\mathsf{b}}$ can break KZG's d-binding or $\mathcal{B}_{\mathsf{ib}}$ can break KZG's d-interpolation binding. $\mathcal{E}_{\mathsf{css}}$ aborts when $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{true}$.

Next, assume $\mathsf{bad}_{\mathsf{ib}} = \mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{false}$. By the above argument $\hat{f}^*(Y)$, $\hat{r}^*(Y)$, and $\bar{\ell}^*(Y)$ are polynomials that interpolate corresponding data on the interval $[\kappa_\delta]$ yet have degree $\leq d$. (The latter follows trivially since $|\mathcal{T}| = d+1$.) Since

$$\kappa_\delta \geq d+1 > \max(\deg(\hat{f}^*), \deg(\hat{r}^*), \deg(\bar{\ell}^*)) \ ,$$

$\hat{f}^*$, $\hat{r}^*$, and $\bar{\ell}^*$ are unique polynomials that evaluate given values at all δ_{k_1}. Since $\mathsf{bad}_{\mathsf{ib}} = \mathsf{false}$, $\hat{f}^*$, $\hat{r}^*$, and $\bar{\ell}^*$ agree with $[\hat{f}(\tau), \hat{r}(\tau), \bar{\ell}(\tau)]_1$. Thus, $\mathcal{E}_{\mathsf{css}}$ has extracted three polynomials that agree with the corresponding commitments and openings and also have small degrees. The proof now proceeds as an AGM proof would, given the polynomials returned by the algebraic adversary as an explanation.

Define

$$G(Y) := \Delta_{\mathsf{y}}(Y)\hat{f}^*(Y) - \left(\Delta_{\mathsf{y}}(Y)(Y^{\mathsf{n}_{\mathsf{y}}} - \mathsf{x})\hat{r}^*(Y) + (Y-\mathsf{y})\bar{\ell}^*(Y) + \vartheta\Delta_{\mathsf{y}}(Y)\right) \ .$$

By Eq. (2) and the argument that resulted in $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{false}$, G has κ_δ roots δ_{k_1}. Recall $\deg(\Delta_{\mathsf{y}}) = d - (\mathsf{n}_{\mathsf{y}} - 2)$. Since $\deg(G) \leq \max((d - \mathsf{n}_{\mathsf{y}} + 2) + d, (d - \mathsf{n}_{\mathsf{y}} + 2) + \mathsf{n}_{\mathsf{y}} + d, d+1, d - (\mathsf{n}_{\mathsf{y}} - 2)) = 2d+2$ (since $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{false}$, we know that $\deg(\hat{f}^*), \deg(\hat{r}^*), \deg(\bar{\ell}^*) \leq d$) and

$$\kappa_\delta \geq 2d+3 > 2d+2 \ ,$$

$G(Y) \equiv 0$ as a polynomial. Since the rest of $G(Y)$ divides by $\Delta_{\mathsf{y}}(Y)$, so does $(Y-\mathsf{y})\bar{\ell}^*(Y)$. Since $\gcd(Y-\mathsf{y}, \Delta_{\mathsf{y}}(Y)) = 1$, $\bar{\ell}^*(Y) = \Delta_{\mathsf{y}}(Y)\hat{\ell}^*(Y)$ for some polynomial $\hat{\ell}^*$ with $\deg(\hat{\ell}^*) \leq \mathsf{n}_{\mathsf{y}} - 2$.

Dividing $G(Y) \equiv 0$ by $\Delta_{\mathsf{y}}(Y)$, we get

$$\hat{f}^*(Y) = (Y^{\mathsf{n}_{\mathsf{y}}} - \mathsf{x})\hat{r}^*(Y) + (Y-\mathsf{y})\hat{\ell}^*(Y) + \vartheta \ . \tag{6}$$

Writing $\hat{p}^*(Y) = (Y-\mathsf{y})\hat{\ell}^*(Y) + \vartheta$, we get that the conditions of Lemma 1 are satisfied for $\hat{f}^*$. Thus, writing $\hat{f}^*(Y) = \sum_{i=0}^{\mathsf{n}_{\mathsf{y}}-1} f_i(Y^{\mathsf{n}_{\mathsf{y}}})Y^i$, the bivariate polynomial $Q(X,Y) = \sum_{i=0}^{\mathsf{n}_{\mathsf{y}}-1} f_i(X)Y^i$ satisfies $\hat{f}^*(Y) = Q(Y^{\mathsf{n}_{\mathsf{y}}}, Y)$ and $Q(\mathsf{x}, Y) = \hat{p}^*(Y) = (Y-\mathsf{y})\hat{\ell}^*(Y) + \vartheta$. But then, $Q(\mathsf{x}, \mathsf{y}) = \vartheta$.

Thus, $\mathcal{E}_{\mathsf{css}}$ succeeds by returning Q when $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{bad}_{\mathsf{ib}} = \mathsf{false}$. More precisely, $\mathsf{Adv}^{\mathrm{ss}}_{\mathsf{PGen},\mathrm{Orca},\boldsymbol{\kappa},\mathcal{E}_{\mathsf{css}},\mathcal{A}_{\mathsf{css}}}(\lambda) \leq \mathsf{Adv}^{\mathrm{ib}}_{\mathsf{PGen},\mathsf{PC},d,\mathcal{B}_{\mathsf{ib}}}(\lambda) + \mathsf{Adv}^{\mathrm{b}}_{\mathsf{PGen},\mathsf{PC},d,\mathcal{B}_{\mathsf{b}}}(\lambda)$ for some PPT adversaries $\mathcal{B}_{\mathsf{ib}}$ and $\mathcal{B}_{\mathsf{b}}$. □

From [AFK22], it follows that, after Fiat-Shamir, the evaluation protocol of non-interactive Orca is tightly knowledge-sound (and thus, Orca is extractable) in the random oracle model under ARSDH. Critically, in the interactive Orca, σ is sampled after δ and ϱ after σ, so in the non-interactive Orca, one must set $\sigma = \mathsf{H}(\ldots, \delta)$ and $\varrho = \mathsf{H}(\ldots, \delta, \sigma, v_\sigma)$. (In Orca_{agm}, there is no σ or ϱ.)

4 Dolphin: Multilinear Polynomial Commitment Scheme

4.1 Basics of Multilinear Polynomials and Inner Products

Let $n = 2^\mu$ and let $\boldsymbol{X}^\mu = (X_1, \ldots, X_\mu)$. Let $\mathbb{B} = \{0,1\}$ and let $\mathbb{B}^\mu$ be the μ-dimensional Boolean hypercube. Denote $\tilde{\mathsf{eq}}_i(\boldsymbol{X}^\mu) := \tilde{\mathsf{eq}}(\langle i \rangle, \boldsymbol{X}^\mu)$, where $\langle i \rangle$ is the binary decomposition of $i-1$. Note that $\{\tilde{\mathsf{eq}}_i : i \in [n]\}$ is the basis of the vector space $\mathbb{F}_{\le 1}[\boldsymbol{X}^\mu]$ while $\{X^{i-1} : i \in [n]\}$ is the basis of $\mathbb{F}_{\le n-1}[X]$. We sometimes write μ in the superscript, denoting the length of both arguments of $\tilde{\mathsf{eq}}$, like in $\tilde{\mathsf{eq}}^\mu(\boldsymbol{x}, \boldsymbol{y})$. For $\boldsymbol{f} \in \mathbb{F}^n$, let $\tilde{f}(\boldsymbol{X}) = \sum_{i=1}^n f_i \tilde{\mathsf{eq}}_i^\mu(\boldsymbol{X})$ be its multilinear extension and $\hat{f}(X) = \sum_{i=1}^n f_i X^{i-1}$ be its monomial (univariate) extension.

Consider the linear map $\mathsf{ml2ld}_\mu : \mathbb{F}_{\le 1}[\boldsymbol{X}^\mu] \to \mathbb{F}_{\le n-1}[X]$ defined by

$$\mathsf{ml2ld}_\mu(\tilde{\mathsf{eq}}_i^\mu(\boldsymbol{X})) := X^{i-1} \ .$$

Thus,

$$\mathsf{ml2ld}_\mu\left(\textstyle\sum_{i=1}^n f_i \tilde{\mathsf{eq}}_i^\mu(\boldsymbol{X})\right) = \textstyle\sum_{i=1}^n f_i X^{i-1} \ . \tag{7}$$

That is, $\mathsf{ml2ld}_\mu(\tilde{f}) = \hat{f} \in \mathbb{F}_{\le n-1}[X]$ and $\mathsf{ml2ld}_\mu^{-1}(\hat{f}) = \tilde{f}$ map a multilinear extension to a low-degree extension and vice versa.

Inner-Product Argument. The following inner-product argument is similar to [BCC+16,MBKM19]; variants of it were used both in [EG25,GPS25]. Let $\boldsymbol{z} = (z_1, \ldots, z_\mu) \in \mathbb{F}^\mu$ (we later use different values instead of μ). Now,

$$\tilde{f}(\boldsymbol{z}) = \textstyle\sum_{i=1}^n f_i \tilde{\mathsf{eq}}_i(\boldsymbol{z}) = \langle \boldsymbol{f}, \mathsf{tensor}(\boldsymbol{z}) \rangle \ ,$$

where $\boldsymbol{f} = (f_1, \ldots, f_n)$ is the coefficient vector of $\tilde{f}$ and

$$\mathsf{tensor}(\boldsymbol{z}) = (\tilde{\mathsf{eq}}_1(\boldsymbol{z}), \ldots, \tilde{\mathsf{eq}}_n(\boldsymbol{z})) \ . \tag{8}$$

Clearly, $\mathsf{tensor}(\boldsymbol{z})$ is equal to the tensor product

$$\mathsf{tensor}(\boldsymbol{z}) = \left(\begin{smallmatrix} 1-z_1 \\ z_1 \end{smallmatrix}\right) \otimes \cdots \otimes \left(\begin{smallmatrix} 1-z_\mu \\ z_\mu \end{smallmatrix}\right) \ ,$$

where we start tensoring from the left and z_1 is thought of as "the least significant bit" of $\boldsymbol{z}$. For example,

$$\left(\begin{smallmatrix} 1-z_1 \\ z_1 \end{smallmatrix}\right) \otimes \left(\begin{smallmatrix} 1-z_2 \\ z_2 \end{smallmatrix}\right) = \begin{pmatrix} (1-z_1)(1-z_2) \\ (1-z_1)z_2 \\ z_1(1-z_2) \\ z_1 z_2 \end{pmatrix} = \begin{pmatrix} \tilde{\mathsf{eq}}_1(\boldsymbol{z}) \\ \tilde{\mathsf{eq}}_2(\boldsymbol{z}) \\ \tilde{\mathsf{eq}}_3(\boldsymbol{z}) \\ \tilde{\mathsf{eq}}_4(\boldsymbol{z}) \end{pmatrix} \ .$$

For $\boldsymbol{z} \in \mathbb{F}^{\mu}$, let $\hat{\Psi}_{\mu}(X; \boldsymbol{z})$ be the univariate polynomial that has $\mathsf{tensor}(\boldsymbol{z})$ in the *reversed* order as the coefficient vector,

$$\hat{\Psi}_{\mu}(X; \boldsymbol{z}) := \sum_{j=0}^{2^{\mu}-1} \tilde{\mathsf{eq}}_{2^{\mu}-j}(\boldsymbol{z}) X^{j} = \sum_{j=1}^{2^{\mu}} \tilde{\mathsf{eq}}_{j}(\boldsymbol{z}) X^{2^{\mu}-j} \in \mathbb{F}_{\leq 2^{\mu}-1}[X] \ . \tag{9}$$

(One can write $\hat{\Psi}_{\mu}(X; \boldsymbol{z}) = \langle \mathsf{reverse}(\mathsf{tensor}(\boldsymbol{z})), \mathsf{pow}(X) \rangle$.) Crucially, $\hat{\Psi}_{\mu}(X; \boldsymbol{z})$ can be evaluated at any $X = \delta$ in time $O(\mu)$ [BGH19] since

$$\hat{\Psi}_{\mu}(X; \boldsymbol{z}) = \prod_{i=1}^{\mu} (z_i + (1 - z_i) X^{2^{i-1}}) \ . \tag{10}$$

We now state a lemma on verifying a multilinear evaluation via an inner-product argument. However, we do not assume that polynomials like $\hat{f}$ have a small degree. In particular, $\hat{f}$ can have a large degree, but $\boldsymbol{z}$ can be much shorter. (A similar result was also used in [EG25,GPS25].)

When $\hat{f}(X) = \sum_{i=1}^{d} f_i X^{i-1}$ for some d and $s < d$, we denote

$$\mathsf{mod}_s(\hat{f})(X) := \hat{f}(X) \pmod{X^s} = \sum_{i=1}^{s} f_i X^{i-1} \text{ and } \boldsymbol{f}_{[s]} := (f_1, \ldots, f_s) \ .$$

Lemma 2 (Multilinear evaluation via IPA). *Let $n = 2^{\mu}$, $\mathsf{n}_{\mathsf{x}} < n$, and $\mathsf{n}_{\mathsf{x}} = 2^{\mu_{\mathsf{x}}}$. Let $\hat{f} = \sum_{i=1}^{n} f_i X^{i-1} \in \mathbb{F}_{\leq n-1}[X]$, $\boldsymbol{z}_x \in \mathbb{F}^{\mu_{\mathsf{x}}}$, and $\vartheta \in \mathbb{F}$. Let $\hat{F} := \mathsf{mod}_{\mathsf{n}_{\mathsf{x}}}(\hat{f})$. Then, the following two claims are equivalent.*

1. *$\tilde{F}(\boldsymbol{z}_x) = \vartheta$, where $\tilde{F}(\boldsymbol{X}^{\mu_{\mathsf{x}}}) = \mathsf{ml2ld}_{\mu_{\mathsf{x}}}^{-1}(\hat{F}) = \sum_{i=1}^{\mathsf{n}_{\mathsf{x}}} f_i \tilde{\mathsf{eq}}_i^{\mu_{\mathsf{x}}}(\boldsymbol{X}^{\mu_{\mathsf{x}}})$.*
2. *there exist polynomials $\hat{h}(X) \in \mathbb{F}_{\leq n-1}[X]$ and $\hat{\ell}(X) \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-2}[X]$, such that*

$$\hat{f}(X) \cdot \hat{\Psi}_{\mu_{\mathsf{x}}}(X; \boldsymbol{z}_x) = \hat{h}(X) X^{\mathsf{n}_{\mathsf{x}}} + \vartheta X^{\mathsf{n}_{\mathsf{x}}-1} + \hat{\ell}(X) \ . \tag{11}$$

Assuming either claim holds and $\hat{f} \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-1}[X]$ (this will hold in the honest case), both $\hat{h}$ and $\hat{\ell}$ can be computed in $O(\mathsf{n}_{\mathsf{x}} \log \mathsf{n}_{\mathsf{x}})$ field multiplications without using FFT.

Proof. ($\Rightarrow$) The coefficient of $X^{\mathsf{n}_{\mathsf{x}}-1}$ in

$$\begin{aligned} \hat{f}(X) \cdot \hat{\Psi}_{\mu_{\mathsf{x}}}(X; \boldsymbol{z}_x) &= \left(\sum_{i=1}^{n} f_i X^{i-1}\right) \left(\sum_{j=1}^{\mathsf{n}_{\mathsf{x}}} \tilde{\mathsf{eq}}_j^{\mu_{\mathsf{x}}}(\boldsymbol{z}_x) X^{\mathsf{n}_{\mathsf{x}}-j}\right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{\mathsf{n}_{\mathsf{x}}} f_i \cdot \tilde{\mathsf{eq}}_j^{\mu_{\mathsf{x}}}(\boldsymbol{z}_x) X^{\mathsf{n}_{\mathsf{x}}-1+(i-j)} \end{aligned} \tag{12}$$

is $\tilde{F}(\boldsymbol{z}_x) = \langle \boldsymbol{f}_{[\mathsf{n}_{\mathsf{x}}]}, \mathsf{tensor}(\boldsymbol{z}_x) \rangle = \sum_{i=1}^{\mathsf{n}_{\mathsf{x}}} f_i \tilde{\mathsf{eq}}_i^{\mu_{\mathsf{x}}}(\boldsymbol{z}_x)$. Thus, $\tilde{F}(\boldsymbol{z}_x) = \vartheta$ if and only if there exist $\hat{h}(X) \in \mathbb{F}_{\leq n-1}[X]$ and $\hat{\ell}(X) \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-2}[X]$ such that Eq. (11) holds. For the degree bound of $\hat{h}(X)$, recall that $\hat{\Psi}_{\mu_{\mathsf{x}}}(X; \boldsymbol{z}_x) \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-1}[X]$.

($\Leftarrow$) Since the coefficient of $X^{\mathsf{n}_{\mathsf{x}}-1}$ in Eq. (12) is $\tilde{F}(\boldsymbol{z}_x)$, we get from Eq. (11) and $\hat{\ell} \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-2}[X]$ that $\tilde{F}(\boldsymbol{z}_x) = \vartheta$.

Finally, assume $\hat{f} \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-1}[X]$; then also $\hat{h} \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-2}[X]$. One can use Eq. (10) to compute $\hat{f}(X) \cdot \hat{\Psi}_{\mu_{\mathsf{x}}}(X; \boldsymbol{z}_x)$ in $O(\mathsf{n}_{\mathsf{x}} \log \mathsf{n}_{\mathsf{x}})$ field multiplications *without using FFT* by successive multiplication of $\hat{f}(X)$ with factors $z_i + (1 - z_i) X^{2^{i-1}}$ of $\hat{\Psi}_{\mu_{\mathsf{x}}}$ and use this to compute $\hat{h} \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-2}[X]$ and $\hat{\ell} \in \mathbb{F}_{\leq \mathsf{n}_{\mathsf{x}}-2}[X]$ by collecting the correct coefficients of $\hat{f}(X) \cdot \hat{\Psi}_{\mu_{\mathsf{x}}}(X; \boldsymbol{z}_x)$. □

4.2 Details of Dolphin

Ganesh, Patranabis, and Singh [GPS25] introduced SamaritanPCS, and independently, Eagen and Gabizon [EG25] introduced Mercury. Both schemes are KZG-based multilinear polynomial commitments with linear prover computation and a constant-size argument. Although they share the same core principles, they employ different (incomparable) optimizations. We follow SamaritanPCS's exposition; most notably, SamaritanPCS explicitly uses an intermediate bivariate polynomial commitment scheme. Since SamaritanPCS is also more efficient than Mercury (see Table 2), our new scheme, Dolphin, builds on SamaritanPCS. We integrate optimizations and de-optimizations to achieve computational special soundness in the standard model while improving on efficiency. We briefly summarize the changes compared to SamaritanPCS (and Mercury):

Optimization. SamaritanPCS permits the commitment key to include powers $[\tau^i]_1$ for $i > d = n - 1$. It then adds degree checks to prove all polynomials (including the committed polynomial) have degree $\leq d$. We remove this generalization, following Mercury. In addition, SamaritanPCS considers two parameters, n_x and n_y, related to the bivariate commitment scheme, such that $n = \mathsf{n}_\mathsf{x} \cdot \mathsf{n}_\mathsf{y}$. Following Mercury, we fix $\mathsf{n}_\mathsf{x} = \mathsf{n}_\mathsf{y}$, which allows to perform additional optimizations. It is also close to the optimal setting of n_x and n_y. This enables us to further optimize the polynomial commitment scheme.

On the other hand, we include several SamaritanPCS optimizations that are not used in Mercury. We use an IPA-based opening of one of the polynomial commitments, which can be batched with the existing IPA in SamaritanPCS. This reduces the number of distinct evaluation points by one compared to Mercury. We utilize another SamaritanPCS optimization to eliminate an additional evaluation point. More precisely, Mercury symmetrizes Lemma 2 by considering an equation like $\hat{f}(X) \cdot (\hat{\Psi}_\mu(X; \boldsymbol{z}) + \hat{\Psi}_{\mu_\mathsf{x}}(X^{-1}; \boldsymbol{z}))$. While this allows the use of a single polynomial instead of two ($\hat{h}(X)$ and $\hat{\ell}(X)$ from Lemma 2), it requires the prover to open some polynomials at two different points. (See [EG25] for details.)

Deoptimization. To prove computational special soundness, we need to de-optimize SamaritanPCS. First, rather than committing to a random linear combination of certain polynomials as in [GPS25], we commit to each polynomial individually. We then use the LinTrick techniques slightly differently compared to the SamaritanPCS. Second, to prove CSS in the standard model, we employ the sanitization technique from [LPS25], adding a single field element to the communication to batch-open all polynomials that would otherwise remain unopened.

We assume KZG is the underlying univariate polynomial commitment scheme. (Ganesh, Patranabis, and Singh [GPS25] also proposed a general multilinear polynomial commitment scheme based on any univariate one.)

Fig. 2 depicts Dolphin. We next explain its construction in detail. For readers' convenience, we adopt the notation of [GPS25].

Dolphin's commitment phase. Let $\mathsf{ck} \leftarrow \mathrm{KZG.}\mathsf{KGen}(1^\lambda, d)$ be the KZG commitment key for $d = n - 1$. Recall that for $\boldsymbol{f} \in \mathbb{F}^n$, $\tilde{f}(\boldsymbol{X}) = \sum_{i=1}^n f_i \tilde{\mathsf{eq}}_i^\mu(\boldsymbol{X})$ and $\hat{f}(X) = \sum_{i=1}^n f_i X^{i-1}$. The multilinear commitment of $\tilde{f}$ is the univariate commitment to $\hat{f}$, $[\hat{f}(\tau)]_1 \leftarrow \mathsf{Com}(\mathsf{ck}, \tilde{f}) = \mathrm{KZG.}\mathsf{Com}(\mathsf{ck}, \hat{f}) = \sum_{i=1}^n f_i [\tau^{i-1}]_1$. Unlike the usual use of the KZG commitment, this commitment does not require the prover to perform univariate interpolation. Notably, the prover directly commits to the coefficient vector, meaning that committing to zero coefficients is (essentially) for free and committing to small coefficients is considerably cheaper than committing to large coefficients. Both properties (*not* shared by the KZG itself) are very important in practice [DP25,ST25].

4.3 Dolphin's Evaluation Phase

Let $n = \mathsf{n}_\mathsf{x}^2$, for $\mathsf{n}_\mathsf{x} \in \mathbb{N}$, where $\mathsf{n}_\mathsf{x} > 1$ is a power of two. Let $\mu := \log_2 n$ and $\mu_\mathsf{x} := \log_2 \mathsf{n}_\mathsf{x} = \mu/2$. [6] Note that μ is even. Let $\tilde{f}(\boldsymbol{X}^\mu)$ be the committed multilinear polynomial.

In the evaluation phase, the prover needs to prove $\tilde{f}(\boldsymbol{z}) = \vartheta$ for given $\boldsymbol{z} \in \mathbb{F}^\mu$ and $\vartheta \in \mathbb{F}$. Let $\boldsymbol{z} = (\boldsymbol{z}_x, \boldsymbol{z}_y) \in \mathbb{F}^{\mu_\mathsf{x}} \times \mathbb{F}^{\mu_\mathsf{x}}$.

Round 1. Let the common coefficient vector of $\tilde{f} \in \mathbb{F}_{\leq 1}[\boldsymbol{X}^\mu]$ and $\hat{f} \in \mathbb{F}_{\leq n-1}[Y]$ be $\boldsymbol{f} = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{\mathsf{n}_\mathsf{x}})$, where $\boldsymbol{g}_i \in \mathbb{F}^{\mathsf{n}_\mathsf{x}}$. Following our general notation rules, $\boldsymbol{g}_i$ is the common coefficient vector of $\tilde{g}_i \in \mathbb{F}_{\leq 1}[\boldsymbol{Y}^{\mu_\mathsf{x}}]$ and $\hat{g}_i \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-1}[Y]$. Note that

$$\tilde{g}_i(\boldsymbol{Y}^{\mu_\mathsf{x}}) = \tilde{f}(\langle i \rangle, \boldsymbol{Y}^{\mu_\mathsf{x}}) \ .$$

Clearly,

$$\hat{f}(Y) = \textstyle\sum_{i=1}^{\mathsf{n}_\mathsf{x}} \hat{g}_i(Y) Y^{\mathsf{n}_\mathsf{x}(i-1)}$$

and, for $\boldsymbol{z}_x \in \mathbb{F}^{\mu_\mathsf{x}}$ and $\boldsymbol{z}_y \in \mathbb{F}^{\mu_\mathsf{x}}$,

$$\begin{aligned} \tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) &= \textstyle\sum_{\boldsymbol{\eta} \in \mathbb{B}^{\mu_\mathsf{x}}} \tilde{f}(\boldsymbol{\eta}, \boldsymbol{z}_y) \tilde{\mathsf{eq}}^{\mu_\mathsf{x}}(\boldsymbol{\eta}, \boldsymbol{z}_x) = \sum_{i=1}^{\mathsf{n}_\mathsf{x}} \tilde{f}(\langle i \rangle, \boldsymbol{z}_y) \tilde{\mathsf{eq}}_i^{\mu_\mathsf{x}}(\boldsymbol{z}_x) \\ &= \textstyle\sum_{i=1}^{\mathsf{n}_\mathsf{x}} \tilde{g}_i(\boldsymbol{z}_y) \tilde{\mathsf{eq}}_i^{\mu_\mathsf{x}}(\boldsymbol{z}_x) \ . \end{aligned} \tag{13}$$

Thus, an evaluation of a multilinear polynomial in $\mu = 2\mu_\mathsf{x}$ variables can be expressed as a sum of n_x evaluations of multilinear polynomials in μ_x variables.

Let

$$\begin{aligned} v_i &:= \tilde{g}_i(\boldsymbol{z}_y) = \tilde{f}(\langle i \rangle, \boldsymbol{z}_y) \ , \\ \hat{v}(X) &:= \textstyle\sum_{i=1}^{\mathsf{n}_\mathsf{x}} v_i X^{i-1} \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-1}[X] \ . \end{aligned}$$

If the prover is honest, then by Eq. (13) and the definition of v_i and $\hat{v}(X)$,

$$\vartheta = \tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \textstyle\sum_{i=1}^{\mathsf{n}_\mathsf{x}} \tilde{g}_i(\boldsymbol{z}_y) \tilde{\mathsf{eq}}_i^{\mu_\mathsf{x}}(\boldsymbol{z}_x) = \sum_{i=1}^{\mathsf{n}_\mathsf{x}} v_i \tilde{\mathsf{eq}}_i^{\mu_\mathsf{x}}(\boldsymbol{z}_x) = \tilde{v}(\boldsymbol{z}_x) \ . \tag{14}$$

This explains Item 1 in Fig. 2. The verifier responds with $\chi \leftarrow\!\!\$\ \mathbb{F}$.

[6] A more general case, where similarly to Section 3, $n = \mathsf{n}_\mathsf{x}\mathsf{n}_\mathsf{y}$, is less efficient.

Round 2. Let

$$Q(X,Y) := \textstyle\sum_{i=1}^{\mathsf{n_x}} X^{i-1}\hat{g}_i(Y) \ .$$

be the unique bivariate polynomial with $\deg_X(Q) \le \mathsf{n_x}-1$ and $\deg_Y(Q) \le \mathsf{n_x}-1$ such that $\hat{f}(Y) = Q(Y^{\mathsf{n_x}}, Y)$. Recall that the Orca bivariate commitment to Q is equal to the KZG univariate commitment to $\hat{f}$ (see Fig. 1).

The prover sets

$$\hat{p}(Y) \leftarrow Q(\chi, Y) = \textstyle\sum_{i=1}^{\mathsf{n_x}} \chi^{i-1}\hat{g}_i(Y) \in \mathbb{F}_{\le \mathsf{n_x}-1}[Y] \ .$$

Recalling that $\mathsf{mI2Id}_{\mu_x} : \tilde{f} \mapsto \hat{f}$ is a vector space homomorphism (see Eq. (7)), $\tilde{p}(\boldsymbol{X}) = \sum_{i=1}^{\mathsf{n_x}} \chi^{i-1}\tilde{g}_i(\boldsymbol{X})$. Thus,

$$\tilde{p}(\boldsymbol{z}_y) = \textstyle\sum_{i=1}^{\mathsf{n_x}} \chi^{i-1}\tilde{g}_i(\boldsymbol{z}_y) = \sum_{i=1}^{\mathsf{n_x}} \chi^{i-1} v_i = \hat{v}(\chi) =: v_\chi \ . \tag{15}$$

The above holds when the prover is honest. In Lemma 3, we handle the case of a malicious prover. The lemma is used by the extractor in Theorem 2 that rewinds χ to $\kappa_\chi = \mathsf{n_x}$ distinct values. That $\kappa_\chi = \mathsf{n_x}$ values of χ suffice is crucial for the proof of the computational special soundness. For completeness, we state the lemma in the case $\mathsf{n_x}$ and $\mathsf{n_y}$ are separate parameters and make the proof independent of the above discussion. We also state Lemma 3 in a form that assumes a priori that $\deg(\hat{v}) \le \kappa_\chi - 1$ for an arbitrary $\kappa_\chi \in [\mathsf{n_x}, n]$.. In the proof of Theorem 2, we will apply it to $\hat{V} := \mathsf{mod}_{\mathsf{n_x}}(\hat{v})$, with $\kappa_\chi = \mathsf{n_x}$.

Lemma 3 (Multilinear evaluation condition). *Let $n = \mathsf{n_x}\mathsf{n_y}$, $\mathsf{n_x} = 2^{\mu_x}$, and $\mathsf{n_y} = 2^{\mu_y}$. Let $\kappa_\chi \in [\mathsf{n_x}, n]$. Fix $\hat{f}(Y) \in \mathbb{F}_{\le n-1}[Y]$, $\boldsymbol{z} = (\boldsymbol{z}_x, \boldsymbol{z}_y) \in \mathbb{F}^{\mu_x} \times \mathbb{F}^{\mu_y}$, and $\vartheta \in \mathbb{F}$. Assume that there exist a polynomial $Q(X,Y)$, with $\deg_X(Q) \le \mathsf{n_x} - 1$ and $\deg_Y(Q) \le \mathsf{n_y} - 1$, and a polynomial $\hat{v} \in \mathbb{F}_{\le \kappa_\chi - 1}[X]$, such that*

(1) $\hat{f}(Y) = Q(Y^{\mathsf{n_y}}, Y)$,
(2) For κ_χ distinct values of $\chi \in \mathbb{F}$, $\hat{p}_\chi(Y) := Q(\chi, Y) \in \mathbb{F}_{\le \mathsf{n_y}-1}[Y]$ satisfies $\tilde{p}_\chi(\boldsymbol{z}_y) = \hat{v}(\chi)$. Here, $\tilde{p}_\chi = \mathsf{mI2Id}_{\mu_y}^{-1}(\hat{p}_\chi)$ as usual.
(3) $\tilde{v}(\boldsymbol{z}_x) = \vartheta$ (it follows from the rest of the proof that $\hat{v} \in \mathbb{F}_{\le \mathsf{n_x}-1}[X]$ so $\tilde{v}$ is well-defined).

Then, $\tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \vartheta$.

A similar result is used in [EG25,GPS25]; however, since they are not proving CSS, they give a weaker version of it, only proving that Item 2 holds with a negligible probability over the choice of $\chi \leftarrow\!\!\$\ \mathbb{F}$.

Proof. Write $Q(X,Y) = \sum_{i=1}^{\mathsf{n_x}} X^{i-1}\hat{g}_i(Y)$ with $\hat{g}_i \in \mathbb{F}_{\le \mathsf{n_y}-1}[Y]$ (possible since $\deg_X(Q) \le \mathsf{n_x} - 1$). Define $\tilde{g}_i := \mathsf{mI2Id}_{\mu_y}^{-1}(\hat{g}_i) \in \mathbb{F}_{\le 1}[\boldsymbol{X}^{\mu_y}]$. By Item 1, $\hat{f}(Y) = Q(Y^{\mathsf{n_y}}, Y) = \sum_{i=1}^{\mathsf{n_x}} \hat{g}_i(Y) Y^{\mathsf{n_y}(i-1)}$.

For any $\chi \in \mathbb{F}$, $\hat{p}_\chi(Y) = Q(\chi, Y) = \sum_{i=1}^{\mathsf{n_x}} \chi^{i-1}\hat{g}_i(Y)$. Thus, $\tilde{p}_\chi(\boldsymbol{z}_y) = \sum_{i=1}^{\mathsf{n_x}} \chi^{i-1}\tilde{g}_i(\boldsymbol{z}_y)$. By Item 2, for κ_χ pairwise distinct χ,

$$\hat{v}(\chi) = \tilde{p}_\chi(\boldsymbol{z}_y) = \textstyle\sum_{i=1}^{\mathsf{n_x}} \chi^{i-1}\tilde{g}_i(\boldsymbol{z}_y) \ .$$

We need κ_χ values since $\hat{v}$ is a priori only guaranteed to have degree $\leq \kappa_\chi - 1$; we restrict the degree of $\hat{v}$ next.

Let $\hat{F}(X) := \sum_{i=1}^{\mathsf{n_x}} \tilde{g}_i(\boldsymbol{z}_y) X^{i-1} \in \mathbb{F}_{\leq \mathsf{n_x}-1}[X]$. Then $\hat{v}(\chi) = \hat{F}(\chi)$ for κ_χ distinct χ, while $\deg(\hat{v}) \leq \kappa_\chi - 1$ and $\deg(\hat{F}) \leq \mathsf{n_x} - 1 \leq \kappa_\chi - 1$. (Here, we need $\mathsf{n_x} \leq \kappa_\chi$.) Thus, $\hat{v} = \hat{F}$. In particular, $\hat{v}$'s coefficients satisfy $v_i = \tilde{g}_i(\boldsymbol{z}_y)$ for $1 \leq i \leq \mathsf{n_x}$ and $v_i = 0$ for $i > \mathsf{n_x}$. (Thus, $\deg(\hat{v}) \leq \mathsf{n_x} - 1$.) Equivalently, when writing $Q(X,Y) \in (\mathbb{F}[X])[Y]$,

$$\begin{aligned}\hat{v}(X) &= \mathsf{ml2ld}_{\mu_y}^{-1}(Q(X,Y))(\boldsymbol{z}_y) = \mathsf{ml2ld}_{\mu_y}^{-1}\left(\textstyle\sum_{i=1}^{\mathsf{n_x}} X^{i-1}\hat{g}_i(Y)\right)(\boldsymbol{z}_y) \\ &= \textstyle\sum_{i=1}^{\mathsf{n_x}} X^{i-1}\mathsf{ml2ld}_{\mu_y}^{-1}(\hat{g}_i(Y))(\boldsymbol{z}_y) = \sum_{i=1}^{\mathsf{n_x}} X^{i-1}\tilde{g}_i(\boldsymbol{z}_y) \ .\end{aligned}$$

Using the standard decomposition for multilinear extensions, $\tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \sum_{i=1}^{\mathsf{n_x}} \tilde{g}_i(\boldsymbol{z}_y)\tilde{\mathsf{eq}}_i(\boldsymbol{z}_x)$. Since $v_i = \tilde{g}_i(\boldsymbol{z}_y)$, $\tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \sum_{i=1}^{\mathsf{n_x}} v_i\tilde{\mathsf{eq}}_i(\boldsymbol{z}_x) = \tilde{v}(\boldsymbol{z}_x)$. ($\tilde{v}$ is well-defined since $\deg(\hat{v}) \leq \mathsf{n_x}-1$.) By Item 3, $\tilde{v}(\boldsymbol{z}_x) = \vartheta$. Thus, $\tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \vartheta$. □

Define $\bar{p}(Y) \leftarrow Y^{d-(\mathsf{n_x}-1)}\hat{p}(Y)$; this explains Item 3 in Fig. 2. The verifier responds with $\alpha \leftarrow_\$ \mathbb{F}$.

Round 3. Next, the prover proves that the preconditions of Lemma 3 hold with an overwhelming probability. To ensure that $\hat{p}(Y) = Q(\chi, Y)$, the prover computes the univariate polynomial $\hat{r}(Y)$ satisfying

$$\hat{f}(Y) = (Y^{\mathsf{n_x}} - \chi)\hat{r}(Y) + \hat{p}(Y) \ , \tag{16}$$

where $\hat{p}(Y) \in \mathbb{F}_{\leq \mathsf{n_x}-1}[Y]$. Here, $\hat{r}(Y) \leftarrow R(Y^{\mathsf{n_x}}, Y) \in \mathbb{F}_{\leq d-\mathsf{n_x}}[Y]$ for $R(X,Y) \leftarrow (Q(X,Y) - Q(\chi,Y))/(X-\chi)$.

Recalling $\mathsf{n_x} = 2^{\mu_x}$, define

$$\begin{aligned}\hat{\boldsymbol{\Phi}}_{\mu_x}(X;\chi) &:= \textstyle\sum_{j=0}^{\mathsf{n_x}-1} \chi^{(\mathsf{n_x}-1)-j}X^j = \frac{X^{\mathsf{n_x}}-\chi^{\mathsf{n_x}}}{X-\chi} \\ &= (X+\chi)(X^2+\chi^2)\cdots(X^{2^{\mu_x-1}}+\chi^{2^{\mu_x-1}}) \in \mathbb{F}_{\leq \mathsf{n_x}-1}[X] \ .\end{aligned} \tag{17}$$

(One can write $\hat{\boldsymbol{\Phi}}_{\mu_x}(X;\chi) = \langle \mathsf{reverse}(\mathsf{pow}(\chi)), \mathsf{pow}(X)\rangle$.) Clearly, $\hat{v}(\chi) = v_\chi$ iff the coefficient of $X^{\mathsf{n_x}-1}$ is v_χ in

$$\begin{aligned}\hat{v}(X)\hat{\boldsymbol{\Phi}}_{\mu_x}(X;\chi) &= \left(\textstyle\sum_{i=1}^{\mathsf{n_x}} v_i X^{i-1}\right)\cdot\left(\sum_{j=0}^{\mathsf{n_x}-1}\chi^{(\mathsf{n_x}-1)-j}X^j\right) \\ &= \textstyle\sum_{i=1}^{\mathsf{n_x}}\sum_{j=0}^{\mathsf{n_x}-1} v_i\chi^j X^{(\mathsf{n_x}-1)+(i-j-1)} \ .\end{aligned} \tag{18}$$

We batch the checks $\tilde{v}(\boldsymbol{z}_x) = \vartheta$ and $\tilde{p}(\boldsymbol{z}_y) = \hat{v}(\chi)$ to a single IP check

$$\langle \boldsymbol{v}, \mathsf{tensor}(\boldsymbol{z}_x) + \alpha\cdot\mathsf{pow}(\chi)\rangle - \langle \boldsymbol{p}, \alpha\cdot\mathsf{tensor}(\boldsymbol{z}_y))\rangle = \vartheta$$

(possible since $\boldsymbol{z}_x, \boldsymbol{z}_y \in \mathbb{F}^{\mu_x}$ and $n = \mathsf{n_x}^2$). Here,

$$\mathsf{pow}(\chi) := (1, \ldots, \chi^{\mathsf{n_x}-1})$$

and $\mathsf{tensor}(\boldsymbol{z}_y)$ is as defined in Eq. (8). In the IP protocol, the prover exhibits polynomials $\hat{h}(Y) \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-2}[Y]$ and $\hat{\ell}(Y) \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-2}[Y]$, such that (recall $\mathsf{n}_\mathsf{x} = \mathsf{n}_\mathsf{y}$)

$$\begin{aligned} \hat{v}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_x)+\alpha\cdot(\hat{v}(X)\hat{\Phi}_{\mu_\mathsf{x}}(X;\chi)-\hat{p}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_y)) \\ = \hat{h}(X)X^{\mathsf{n}_\mathsf{x}}+\vartheta X^{\mathsf{n}_\mathsf{x}-1}+\hat{\ell}(X) \in \mathbb{F}_{\leq 2\mathsf{n}_\mathsf{x}-2}[X] \ . \end{aligned} \tag{19}$$

(Recall that in the honest case, $\hat{v}(X) \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-1}[X]$, $\hat{p}(X) \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-1}[X]$, $\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_x) \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-1}[X]$, and $\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_y) \in \mathbb{F}_{\leq \mathsf{n}_\mathsf{x}-1}[X]$. In the malicious case, we will bound the degrees of $\hat{v}$ and $\hat{p}$.)[7] The first term in Eq. (19) follows from Lemma 2 applied to $\tilde{v}(\boldsymbol{z}_x) = \vartheta$. For the second term, the coefficient of $X^{\mathsf{n}_\mathsf{x}-1}$ in $\hat{v}(X)\hat{\Phi}_{\mu_\mathsf{x}}(X;\chi)$ equals $\hat{v}(\chi)$, while the coefficient of $X^{\mathsf{n}_\mathsf{x}-1}$ in $\hat{p}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_y)$ equals $\tilde{p}(\boldsymbol{z}_y)$. Batching the two identities with challenge α yields Eq. (19). Since $\tilde{p}$ and $\tilde{v}$ have size n_x, this takes $O(\mathsf{n}_\mathsf{x}\log\mathsf{n}_\mathsf{x})$ time with $O(1)$ communication.

Let $\bar{\ell}(Y) \leftarrow Y^{d-(\mathsf{n}_\mathsf{x}-2)}\hat{\ell}(Y) \in \mathbb{F}_{\leq d}[Y]$.

This explains Item 5 in Fig. 2. The verifier responds with a uniformly random δ (for the security proof to go through, we need $\delta \neq 0$ and $\delta^{\mathsf{n}_\mathsf{x}} \neq \chi$).

Round 4: sanitization. Not opening a polynomial commitment means one cannot prove computational special soundness in the standard model, [LPS24,LPS25]. Opening all polynomial commitments separately introduces communication overhead. To reduce overhead, we use the sanitization technique of [LPS25]. When using the original sanitization technique of [LPS25], the prover batch-opens all (six in Dolphin) unopened polynomials using a sanitizer challenge. Similarly to Orca (see Section 3), we optimize it by letting the prover batch-open *four* unopened polynomials ($\hat{f}$, $\hat{v}$, $\bar{p}$, and $\hat{h}$) and then opening the other two ($\hat{r}$ and $\bar{\ell}$) by using the linearity of the linearization polynomial. Concretely, after receiving (δ,σ), the prover computes $\upsilon_\sigma \leftarrow \hat{f}(\delta)+\sigma\hat{v}(\delta)+\sigma^2\bar{p}(\delta)+\sigma^3\hat{h}(\delta)$. The prover then batch-opens the remaining polynomials using a different challenge ϱ and the sanitized batch of polynomials. Without sanitization, the extractor would not be able to extract $\hat{f}$, $\hat{v}$, $\bar{p}$, and $\hat{h}$, and thus would not be able to extract the witness. See [LPS25] for background on sanitization.

Next, the prover proves that the key equations Eqs. (16) and (19) hold, scaled to prove the following degree bounds:

- $\deg(Y^{-d+(\mathsf{n}_\mathsf{x}-1)}\bar{p}(Y)) = \deg(\hat{p}) \leq \mathsf{n}_\mathsf{x}-1$,
- $\deg(Y^{-d+(\mathsf{n}_\mathsf{x}-2)}\bar{\ell}(Y)) = \deg(\hat{\ell}) \leq \mathsf{n}_\mathsf{x}-2$,

[7] The part masked with α in Eq. (19) uses the inner product protocol to prove that $\hat{v}(\chi) = \upsilon_\chi$ without KZG-opening $\hat{v}(X)$ at the point χ. The same technique can be generalized to batch-open many polynomial commitments at multiple locations while computing only a single KZG opening. This technique is more efficient than alternative batch-openings [BDFG20] if the upper-level protocol uses an inner-product protocol for other purposes (as in SamaritanPCS and Dolphin).

Next, we define the linearization polynomial

$$\begin{aligned}\hat{L}(X) := &(\delta^{d-(\mathsf{n_x}-2)}\hat{v}(X)(\hat{\Psi}_{\mu_x}(\delta;\boldsymbol{z}_x)+\alpha\hat{\Phi}_{\mu_x}(\delta;\chi))-\alpha\delta\bar{p}(X)\hat{\Psi}_{\mu_x}(\delta;\boldsymbol{z}_y))\\ &-(\delta^{d+2}\hat{h}(X)+\delta^{d+1}\vartheta+\bar{\ell}(X))\\ &+\varrho\cdot(\delta^{d-(\mathsf{n_x}-1)}\hat{f}(X)-\delta^{d-(\mathsf{n_x}-1)}(\delta^{\mathsf{n_x}}-\chi)\hat{r}(X)-\bar{p}(X))\\ &+\varrho^2\cdot(\hat{f}(X)+\sigma\hat{v}(X)+\sigma^2\bar{p}(X)+\sigma^3\hat{h}(X)-\upsilon_\sigma)\\ &=(\varrho\delta^{d-(\mathsf{n_x}-1)}+\varrho^2)\cdot\hat{f}(X)\\ &+(\delta^{d-(\mathsf{n_x}-2)}(\hat{\Psi}_{\mu_x}(\delta;\boldsymbol{z}_x)+\alpha\hat{\Phi}_{\mu_x}(\delta;\chi))+\varrho^2\sigma)\cdot\hat{v}(X)\\ &+(-\alpha\delta\hat{\Psi}_{\mu_x}(\delta;\boldsymbol{z}_y)-\varrho+\varrho^2\sigma^2)\cdot\bar{p}(X)+\left(-\delta^{d+2}+\varrho^2\sigma^3\right)\cdot\hat{h}(X)\\ &-\bar{\ell}(X)-\varrho\delta^{d-(\mathsf{n_x}-1)}(\delta^{\mathsf{n_x}}-\chi)\cdot\hat{r}(X)-(\delta^{d+1}\vartheta+\varrho^2\upsilon_\sigma)\ ,\end{aligned} \tag{20}$$

where the ϱ^2 block adds sanitization (needed only for the standard-model computational special soundness). We scaled $\hat{L}$ by powers of δ so one does not have to invert δ. By Eqs. (16) and (19) and the choice of υ_σ, the coefficient of each ϱ^i of $\hat{L}(X)$ evaluates to 0 at $X=\delta$. Thus, $(X-\delta)\mid\hat{L}(X)$.

$\mathcal{P}_{\mathsf{Eval}}$ computes a KZG evaluation argument $[\hat{H}(\tau)]_1 \leftarrow [\hat{L}(\tau)/(\tau-\delta)]_1$ to prove $\hat{L}(\delta)=0$. Since $(X-\delta)\mid\hat{L}(X)$, the prover can even do that when $\delta=\tau$. (This edge case happens with the probability $O(1/|\mathbb{F}|)$.)

The verifier computes $[\hat{L}(\tau)]_1$ and checks $[\hat{H}(\tau)]_1$ using $[\hat{L}(\tau)]_1$. This explains Items 10 and 11 in Fig. 2.

Efficiency. Dolphin's transcript is $(\mathbb{x}=([\hat{f}(\tau)]_1,\boldsymbol{z},\vartheta),[\hat{v}(\tau)]_1;\chi;[\hat{r}(\tau),\bar{p}(\tau)]_1;\alpha;[\hat{h}(\tau),\bar{\ell}(\tau)]_1;\delta;\sigma;\upsilon_\sigma;\varrho;[\hat{H}(\tau)]_1)$. (This also includes the common input $\mathbb{x}$.) We do not include the verifier's messages in the computation costs, as they are replaced by random oracle outputs after the use of Fiat-Shamir [FS87].

Critically, in the interactive Dolphin (see Fig. 2), σ is sampled after δ and ϱ is sampled after σ, so in the non-interactive Dolphin, $\sigma=\mathsf{H}(\ldots,\delta)$ and $\varrho=\mathsf{H}(\ldots,\delta,\sigma,\upsilon_\sigma)$. In the non-interactive Dolphin$_{agm}$ (no σ), $\varrho=\mathsf{H}(\ldots,\delta)$. *This order is security-critical.*

To see the prover's computation, note that

- Committing to $\hat{v}(X)$ takes an $\mathsf{n_x}$-wide MSM.
- Committing to $\bar{p}(Y)$ takes an $\mathsf{n_x}$-wide MSM.
- Committing to $\hat{r}(X)$ takes an $(n-\mathsf{n_x})$-wide MSM.
- Committing to $\hat{h}(Y)$ takes an $(\mathsf{n_x}-1)$-wide MSM.
- Committing to $\bar{\ell}(Y)$ takes an $(\mathsf{n_x}-1)$-wide MSM. We count MSM width by the number of non-zero coefficients; the shift by a power of Y makes $\bar{\ell}$ and $\bar{p}$ sparse with at most $\mathsf{n_x}-1$ and $\mathsf{n_x}$ non-zeros.
- Computing $[\hat{H}(\tau)]_1$ takes an $(n-1)$-wide MSM.

The first operation (committing to $\hat{v}(X)$) is the only one where the polynomial can have small coefficients. All other polynomials have random coefficients.

$\mathsf{KGen}(1^\lambda, \mu, \mathbf{1})$**:** return $(\mathsf{ck}, \mathsf{tk}) \leftarrow \mathrm{KZG.KGen}(1^\lambda, d)$. // $n = 2^\mu$, $d = n-1$, $n = \mathsf{n}_\times^2$
$\mathsf{Com}(\mathsf{ck}, \tilde{f} \in \mathbb{F}_{\le 1}[\boldsymbol{X}^\mu])$**:** decompose coefficient vector $\boldsymbol{f} \in \mathbb{F}^n$ of $\tilde{f}$ as $(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{\mathsf{n}_\times})$ with $\boldsymbol{g}_i \in \mathbb{F}^{\mathsf{n}_\times}$. $Q(X,Y) \leftarrow \sum_{i=1}^{\mathsf{n}_\times} X^{i-1} \hat{g}_i(Y)$.
$[\hat{f}(\tau)]_1 \leftarrow \mathrm{KZG.Com}(\mathsf{ck}, \hat{f}) = \sum_{i=1}^{n} f_i [\tau^{i-1}]_1$. Return $([\hat{f}(\tau)]_1, \mathsf{dc} = \hat{f})$.
Evaluation $\langle \mathcal{P}_{\mathsf{Eval}}(\tilde{f}) \leftrightarrow \mathcal{V}_{\mathsf{Eval}} \rangle([\hat{f}(\tau)]_1, \boldsymbol{z} \in \mathbb{F}^\mu, \vartheta \in \mathbb{F})$**:** // $\tilde{f}(\boldsymbol{z}) = \vartheta$

1. $\mathcal{P}_{\mathsf{Eval}}$: Let $\boldsymbol{z} = (\boldsymbol{z}_x, \boldsymbol{z}_y)$ with $\boldsymbol{z}_x \in \mathbb{F}^{\mu_\times}$, $\boldsymbol{z}_y \in \mathbb{F}^{\mu_\times}$.
   For $i \in [\mathsf{n}_\times]$: $v_i \leftarrow \tilde{g}_i(\boldsymbol{z}_y) = \tilde{f}(\langle i \rangle, \boldsymbol{z}_y)$.
   $\hat{v}(X) \leftarrow \sum_{i=1}^{\mathsf{n}_\times} v_i X^{i-1} \in \mathbb{F}_{\le \mathsf{n}_\times - 1}[X]$.
   Send $[\hat{v}(\tau)]_1 \leftarrow \mathrm{KZG.Com}(\mathsf{ck}, \hat{v} \in \mathbb{F}_{\le \mathsf{n}_\times - 1}[X])$.
2. $\mathcal{V}_{\mathsf{Eval}}$: send uniformly random $\chi \leftarrow_\$ \mathbb{F}$.

---

3. $\mathcal{P}_{\mathsf{Eval}}$: $\hat{p}(Y) \leftarrow Q(\chi, Y) \in \mathbb{F}_{\le \mathsf{n}_\times - 1}[Y]$. $\bar{p}(Y) \leftarrow Y^{d-(\mathsf{n}_\times - 1)} \hat{p}(Y) \in \mathbb{F}_{\le d}[Y]$.
   $R(X,Y) \leftarrow (Q(X,Y) - Q(\chi, Y))/(X - \chi)$.
   $\hat{r}(Y) \leftarrow R(Y^{\mathsf{n}_\times}, Y) \in \mathbb{F}_{\le d - \mathsf{n}_\times}[Y]$.
   $[\hat{r}(\tau)]_1 \leftarrow \mathrm{KZG.Com}(\mathsf{ck}, \hat{r} \in \mathbb{F}_{\le d - \mathsf{n}_\times}[Y])$. // **expensive computation**
   $[\bar{p}(\tau)]_1 \leftarrow \mathrm{KZG.Com}(\mathsf{ck}, \bar{p} \in \mathbb{F}_{\le d}[Y])$.
   Send $[\hat{r}(\tau), \bar{p}(\tau)]_1$.
4. $\mathcal{V}_{\mathsf{Eval}}$: send uniformly random $\alpha \leftarrow_\$ \mathbb{F}$.

---

5. $\mathcal{P}_{\mathsf{Eval}}$: Compute $\hat{\ell}(X), \hat{h}(X) \in \mathbb{F}_{\le \mathsf{n}_\times - 2}[X]$, such that $\hat{v}(X)(\hat{\Psi}_{\mu_\times}(X; \boldsymbol{z}_x) + \alpha \hat{\Phi}_{\mu_\times}(X; \chi)) - \alpha \hat{p}(X) \hat{\Psi}_{\mu_\times}(X; \boldsymbol{z}_y) = \hat{h}(X) X^{\mathsf{n}_\times} + \vartheta X^{\mathsf{n}_\times - 1} + \hat{\ell}(X)$.
   $\bar{\ell}(Y) \leftarrow Y^{d-(\mathsf{n}_\times - 2)} \hat{\ell}(Y) \in \mathbb{F}_{\le d}[Y]$.
   $[\hat{h}(\tau)]_1 \leftarrow \mathrm{KZG.Com}(\mathsf{ck}, \hat{h} \in \mathbb{F}_{\le \mathsf{n}_\times - 2}[X])$.
   $[\bar{\ell}(\tau)]_1 \leftarrow \mathrm{KZG.Com}(\mathsf{ck}, \bar{\ell} \in \mathbb{F}_{\le d}[X])$. // Up to $\mathsf{n}_\times - 1$ non-zero coefficients
   Send $[\hat{h}(\tau), \bar{\ell}(\tau)]_1$.
6. $\mathcal{V}_{\mathsf{Eval}}$: sample $\delta \leftarrow_\$ \mathbb{F}^* \setminus \{t : t^{\mathsf{n}_\times} = \chi\}$. Send δ.

---

7. $\mathcal{V}_{\mathsf{Eval}}$: sample $\sigma \leftarrow_\$ \mathbb{F}$. // For CSS/Fiat-Shamir proof, σ is sent after δ

---

8. $\mathcal{P}_{\mathsf{Eval}}$: send $\upsilon_\sigma \leftarrow \hat{f}(\delta) + \sigma \hat{v}(\delta) + \sigma^2 \bar{p}(\delta) + \sigma^3 \hat{h}(\delta)$.
9. $\mathcal{V}_{\mathsf{Eval}}$: sample $\varrho \leftarrow_\$ \mathbb{F}$. Send ϱ.

---

10. $\mathcal{P}_{\mathsf{Eval}}$: Compute $\hat{L}(X) = \sum_{i=0}^{d} L_i X^i$ as per Eq. (20). $[\hat{L}(\tau)]_1 \leftarrow \sum_{i=0}^{d} L_i [\tau^i]_1$.
    $(\ldots, [\hat{\Pi}(\tau)]_1) \leftarrow \mathrm{KZG.Eval}(\mathsf{ck}, [\hat{L}(\tau)]_1, \delta, \hat{L})$. // The first output is ignored
    Send $[\hat{\Pi}(\tau)]_1$. // $[\hat{\Pi}(\tau)]_1$**: expensive comp.**
11. $\mathcal{V}_{\mathsf{Eval}}$: check $\mathrm{KZG.Vf}(\mathsf{ck}, [\hat{L}(\tau)]_1, \delta, 0, [\hat{\Pi}(\tau)]_1)$ by checking

$$\left(\begin{array}{r} (\varrho\delta^{d-(\mathsf{n}_\times-1)} + \varrho^2)[\hat{f}(\tau)]_1 \\ +(\delta^{d-(\mathsf{n}_\times-2)}(\hat{\Psi}_{\mu_\times}(\delta; \boldsymbol{z}_x) + \alpha \hat{\Phi}_{\mu_\times}(\delta; \chi)) + \varrho^2\sigma)[\hat{v}(\tau)]_1 \\ +(-\alpha\delta \hat{\Psi}_{\mu_\times}(\delta; \boldsymbol{z}_y) - \varrho + \varrho^2\sigma^2)[\bar{p}(\tau)]_1 \\ +(-\delta^{d+2} + \varrho^2\sigma^3)[\hat{h}(\tau)]_1 - [\bar{\ell}(\tau)]_1 \\ -\varrho\delta^{d-(\mathsf{n}_\times-1)}(\delta^{\mathsf{n}_\times} - \chi)[\hat{r}(\tau)]_1 \\ -(\delta^{d+1}\vartheta + \varrho^2\upsilon_\sigma)[1]_1 \\ +\delta[\hat{\Pi}(\tau)]_1 \end{array}\right) \bullet [1]_2 = [\hat{\Pi}(\tau)]_1 \bullet [\tau]_2 \ . \tag{21}$$

Fig. 2. Dolphin multilinear commitment scheme. The highlighted (blue) part of the protocol is needed to prove computational special soundness in the standard model.

Assuming that one SM takes a unit cost in each MSM (this does not hold when one uses Pippenger's algorithm, but gives a rough estimate), we get that there are $2n + 3\mathsf{n}_\mathsf{x} - 3 = 2n + 3\sqrt{n} - 3$ scalar multiplications.

The verifier's computation is dominated by a 7-wide MSM in $\mathbb{G}_1$ (note that $[\bar{\ell}(\tau)]_1$ is not multiplied by a scalar) and a two-wide multipairing. In addition, the verifier executes $O(\log n)$ $\mathbb{F}$-operations. Table 2 summarizes the efficiency of Mercury, SamaritanPCS, and Dolphin.

Sending and verifying υ_σ is only needed for sanitization, which helps with computational special soundness. If we are satisfied with proving knowledge soundness in the AGM, we can omit the field element υ_σ, making the argument 32 bytes (when using the BLS12-381 curve [BGM17]) shorter.

4.4 Computational Special Soundness

Ganesh, Patranabis, and Singh [GPS25] claimed but did not prove that if d-PDL (see Definition 1) is hard, the interactive evaluation protocol of SamaritanPCS has knowledge soundness in the algebraic group model. Eagen and Gabizon [EG25] claimed and proved the same result for Mercury. While one can obtain a non-interactive knowledge sound argument from an interactive one after using Fiat-Shamir, the non-interactive argument can have non-tight security. We prove that Dolphin has computational special soundness in the standard model under ARSDH [LPS24]. From [AFK22], it follows that, after Fiat-Shamir, the evaluation protocol of non-interactive Dolphin is tightly knowledge-sound (and thus, Dolphin is extractable) in the random oracle model under ARSDH.

Theorem 2. *Let $d = n - 1$, $n = \mathsf{n}_\mathsf{x}^2$. Assume $n/|\mathbb{F}|$ is negligible.*[8] *Assume KZG is binding and interpolation binding; both hold under the (d-)ARSDH assumption. Let $\boldsymbol{\kappa} = (\kappa_\chi = \mathsf{n}_\mathsf{x}, \kappa_\alpha = 2, \kappa_\delta = 2d + 3, \kappa_\sigma = 4, \kappa_\varrho = 3)$ and $\boldsymbol{N} = (|\mathbb{F}|, |\mathbb{F}|, N_\delta, |\mathbb{F}|, |\mathbb{F}|)$ where $N_\delta = \min_{\chi \in \mathbb{F}} |\mathbb{F}^* \setminus \{t : t^{\mathsf{n}_\mathsf{x}} = \chi\}| \geq |\mathbb{F}| - (\mathsf{n}_\mathsf{x} + 1)$. Interactive CSS-version of Dolphin has $\boldsymbol{\kappa}$-out-of-$\boldsymbol{N}$ computational special soundness in the standard model.*

Note that evaluation binding is not required and that the tree size is $\Theta(n^{1.5})$.

Proof. Denote $\boldsymbol{k}_j := (k_1, \ldots, k_j)$. In the $\boldsymbol{\kappa}$-computational special-soundness proof of Dolphin for a fixed $\boldsymbol{z}$ and ϑ, we construct a KZG's binding adversary $\mathcal{B}_\mathsf{b}$, interpolation-binding adversary $\mathcal{B}_\mathsf{ib}$, and a multilinear polynomial commitment scheme computational special-soundness extractor $\mathcal{E}_\mathsf{css}$. $\mathcal{B}_\mathsf{b}$ and $\mathcal{B}_\mathsf{ib}$ invoke a computational special-soundness adversary $\mathcal{A}_\mathsf{css}$ to obtain an accepting $\boldsymbol{\kappa}$-transcript tree $\mathsf{Tree} = (\mathsf{tr}_{\boldsymbol{k}_5})_{\boldsymbol{k}_5 \in [\kappa_\chi] \times [\kappa_\alpha] \times [\kappa_\delta] \times [\kappa_\sigma] \times [\kappa_\varrho]}$, while $\mathcal{E}_\mathsf{css}$ obtains Tree as an input (but it is still created by $\mathcal{A}_\mathsf{css}$). Here,

$$\mathsf{tr}_{\boldsymbol{k}_5} = \begin{pmatrix} [\hat{f}(\tau)]_1, \boldsymbol{z}, \vartheta; [\hat{v}(\tau)]_1; \chi_{k_1}; [\hat{r}_{k_1}(\tau), \bar{p}_{k_1}(\tau)]_1; \alpha_{\boldsymbol{k}_2}; \\ [\hat{h}_{\boldsymbol{k}_2}(\tau), \bar{\ell}_{\boldsymbol{k}_2}(\tau)]_1; \delta_{\boldsymbol{k}_3}; \sigma_{\boldsymbol{k}_4}; \upsilon_{\sigma, \boldsymbol{k}_4}; \varrho_{\boldsymbol{k}_5}; [\hat{\Pi}_{\boldsymbol{k}_5}(\tau)]_1 \end{pmatrix} .$$

[8] More precisely, we need $|\mathbb{F}| \geq 2n + \mathsf{n}_\mathsf{x} + 3$ for Tree to exist. For the knowledge error to be negligible, we need $n/|\mathbb{F}|$ to be negligible.

(Crucially, we rewind σ after rewinding δ.) Denote

$$\begin{aligned}
[g_{0,\boldsymbol{k}_3}(\tau)]_1 &:= [(\delta_{\boldsymbol{k}_3}^{d-(\mathsf{n}_\mathsf{x}-2)}(\hat{\Psi}_{\mu_\mathsf{x}}(\delta_{\boldsymbol{k}_3};\boldsymbol{z}_x)+\alpha_{\boldsymbol{k}_2}\hat{\Phi}_{\mu_\mathsf{x}}(\delta_{\boldsymbol{k}_3};\chi_{k_1})))\cdot\hat{v}(\tau)]_1 \\
&\quad + [-\alpha_{\boldsymbol{k}_2}\delta_{\boldsymbol{k}_3}\hat{\Psi}_{\mu_\mathsf{x}}(\delta_{\boldsymbol{k}_3};\boldsymbol{z}_y)\cdot\bar{p}_{k_1}(\tau) - \delta_{\boldsymbol{k}_3}^{d+2}\cdot\hat{h}_{\boldsymbol{k}_2}(\tau) - \bar{\ell}_{\boldsymbol{k}_2}(\tau) - \delta_{\boldsymbol{k}_3}^{d+1}\cdot\vartheta]_1 \ , \\
[g_{1,\boldsymbol{k}_3}(\tau)]_1 &:= [\delta_{\boldsymbol{k}_3}^{d-(\mathsf{n}_\mathsf{x}-1)}\cdot\hat{f}(\tau) - \delta_{\boldsymbol{k}_3}^{d-(\mathsf{n}_\mathsf{x}-1)}(\delta_{\boldsymbol{k}_3}^{\mathsf{n}_\mathsf{x}}-\chi_{k_1})\cdot\hat{r}_{k_1}(\tau) - \bar{p}_{k_1}(\tau)]_1 \ , \\
[g_{2,\boldsymbol{k}_4}(\tau)]_1 &:= [\hat{f}(\tau)+\sigma_{\boldsymbol{k}_4}\cdot\hat{v}(\tau)+\sigma_{\boldsymbol{k}_4}^2\cdot\bar{p}_{k_1}(\tau)+\sigma_{\boldsymbol{k}_4}^3\cdot\hat{h}_{\boldsymbol{k}_2}(\tau)-\upsilon_{\sigma,\boldsymbol{k}_4}]_1 \ .
\end{aligned}$$

Since Tree is accepting, by the verifier equation Eq. (21),

$$\left[g_{0,\boldsymbol{k}_3}(\tau)+\varrho_{\boldsymbol{k}_5}g_{1,\boldsymbol{k}_3}(\tau)+\varrho_{\boldsymbol{k}_5}^2g_{2,\boldsymbol{k}_4}(\tau)\right]_1 \bullet [1]_2 = [\hat{\Pi}_{\boldsymbol{k}_5}(\tau)]_1 \bullet [\tau-\delta_{\boldsymbol{k}_3}]_2 \tag{22}$$

for all $\boldsymbol{k}_5$. By the definition of a transcript tree, $\chi_{k_1}\neq\chi_{k_1'}$ for $k_1\neq k_1'$, $\alpha_{k_1k_2}\neq\alpha_{k_1k_2'}$ for any k_1 and $k_2\neq k_2'$, etc.

The verifier equation is a degree-2 polynomial in ϱ, thus $\kappa_\varrho=3$ distinct ϱ values suffice to isolate all coefficients. For any fixed $\boldsymbol{k}_4$, let $\boldsymbol{R}_{\boldsymbol{k}_4}=\mathsf{Van}(\varrho_{\boldsymbol{k}_4 1},\ldots,\varrho_{\boldsymbol{k}_4 3})\in\mathbb{F}^{3\times3}$ be a Vandermonde matrix. Expanding Eq. (22) at the 3 distinct $\varrho_{\boldsymbol{k}_4 j}$'s, we get

$$\boldsymbol{R}_{\boldsymbol{k}_4}\begin{bmatrix} g_{0,\boldsymbol{k}_3}(\tau)\\ g_{1,\boldsymbol{k}_3}(\tau)\\ g_{2,\boldsymbol{k}_4}(\tau)\end{bmatrix}_1 \bullet [1]_2 = \begin{bmatrix}\hat{\Pi}_{\boldsymbol{k}_4 1}\\ \hat{\Pi}_{\boldsymbol{k}_4 2}\\ \hat{\Pi}_{\boldsymbol{k}_4 3}\end{bmatrix}_1 \bullet [\tau-\delta_{\boldsymbol{k}_3}]_2 \ .$$

By the definition of a transcript tree, $\det\boldsymbol{R}_{\boldsymbol{k}_4}\neq0$. Thus, by KZG transcript-homomorphism we can invert the last linear equation system and obtain, for each fixed $\boldsymbol{k}_4$, accepting KZG openings

$$\begin{bmatrix} g_{0,\boldsymbol{k}_3}(\tau)\\ g_{1,\boldsymbol{k}_3}(\tau)\\ g_{2,\boldsymbol{k}_4}(\tau)\end{bmatrix}_1 \bullet [1]_2 = \begin{bmatrix}\hat{\Pi}'_{\boldsymbol{k}_4 1}\\ \hat{\Pi}'_{\boldsymbol{k}_4 2}\\ \hat{\Pi}'_{\boldsymbol{k}_4 3}\end{bmatrix}_1 \bullet [\tau-\delta_{\boldsymbol{k}_3}]_2 \quad \text{for} \quad \begin{bmatrix}\hat{\Pi}'_{\boldsymbol{k}_4 1}\\ \hat{\Pi}'_{\boldsymbol{k}_4 2}\\ \hat{\Pi}'_{\boldsymbol{k}_4 3}\end{bmatrix}_1 := \boldsymbol{R}_{\boldsymbol{k}_4}^{-1}\begin{bmatrix}\hat{\Pi}_{\boldsymbol{k}_4 1}\\ \hat{\Pi}_{\boldsymbol{k}_4 2}\\ \hat{\Pi}_{\boldsymbol{k}_4 3}\end{bmatrix}_1 \tag{23}$$

at $\delta_{\boldsymbol{k}_3}$ of $[g_{0,\boldsymbol{k}_3}(\tau),g_{1,\boldsymbol{k}_3}(\tau),g_{2,\boldsymbol{k}_4}(\tau)]_1$.

Recall $\kappa_\sigma=4$. Next, for any fixed $\boldsymbol{k}_3$, let $\boldsymbol{S}_{\boldsymbol{k}_3}=\mathsf{Van}(\sigma_{\boldsymbol{k}_3 1},\ldots,\sigma_{\boldsymbol{k}_3 4})\in\mathbb{F}^{4\times4}$ be a Vandermonde matrix. Rewrite the last row of Eq. (23) for four different values of the last branch and taking into account the definition of $[g_{2,\boldsymbol{k}_3 j}(\tau)]_1$,

$$\left(\boldsymbol{S}_{\boldsymbol{k}_3}\begin{bmatrix}\hat{f}(\tau)\\ \hat{v}(\tau)\\ \bar{p}_{k_1}(\tau)\\ \hat{h}_{\boldsymbol{k}_2}(\tau)\end{bmatrix}_1 - \begin{pmatrix}\upsilon_{\sigma,\boldsymbol{k}_3 1}\\ \vdots\\ \upsilon_{\sigma,\boldsymbol{k}_3 4}\end{pmatrix}[1]_1\right)\bullet[1]_2 = \begin{bmatrix}\hat{\Pi}'_{\boldsymbol{k}_3 13}\\ \vdots\\ \hat{\Pi}'_{\boldsymbol{k}_3 43}\end{bmatrix}_1 \bullet [\tau-\delta_{\boldsymbol{k}_3}]_2 \ .$$

By the definition of a transcript tree, $\det\boldsymbol{S}_{\boldsymbol{k}_3}\neq0$. Multiplying by $\boldsymbol{S}_{\boldsymbol{k}_3}^{-1}$, we obtain accepting KZG transcripts certifying that $[\hat{f}(\tau)]_1$, $[\hat{v}(\tau)]_1$, $[\bar{p}_{k_1}(\tau)]_1$, and $[\hat{h}_{\boldsymbol{k}_2}(\tau)]_1$ open at $\delta_{\boldsymbol{k}_3}$ to some scalars $\hat{f}(\delta_{\boldsymbol{k}_3})$, $\hat{v}(\delta_{\boldsymbol{k}_3})$, $\bar{p}_{k_1}(\delta_{\boldsymbol{k}_3})$, and $\hat{h}_{\boldsymbol{k}_2}(\delta_{\boldsymbol{k}_3})$. We will show later that the openings do not depend on $\boldsymbol{k}_3$.

$\mathcal{B}_{\mathsf{b}}$, $\mathcal{B}_{\mathsf{ib}}$, and $\mathcal{E}_{\mathsf{css}}$ interpolate the following polynomials for an arbitrary subset $\mathcal{T}\subset[\kappa_\delta]$ of size $d+1$ (we will argue later why *any fixed*, say, k_2 suffices):

1. For *any fixed* $\boldsymbol{k}_2\in[\kappa_\chi]\times[\kappa_\alpha]$: $\hat{f}^*(Y)$ from $(\delta_{\boldsymbol{k}_3},\hat{f}(\delta_{\boldsymbol{k}_3}))_{k_3\in\mathcal{T}}$.

2. For *any fixed* $\boldsymbol{k}_2 \in [\kappa_\chi] \times [\kappa_\alpha]$: $\hat{v}^*(X)$ from $(\delta_{\boldsymbol{k}_3}, \hat{v}(\delta_{\boldsymbol{k}_3}))_{k_3 \in \mathcal{T}}$.
3. For any k_1 and *any fixed* $k_2 \in [\kappa_\alpha]$: $\bar{p}^*_{k_1}(Y)$ from $(\delta_{\boldsymbol{k}_3}, \bar{p}_{k_1}(\delta_{\boldsymbol{k}_3}))_{k_3 \in \mathcal{T}}$.
4. For any $\boldsymbol{k}_2$: $\hat{h}^*_{\boldsymbol{k}_2}(Y)$ from $(\delta_{\boldsymbol{k}_3}, \hat{h}(\delta_{\boldsymbol{k}_3}))_{k_3 \in \mathcal{T}}$.
5. For any $\boldsymbol{k}_2$: $\bar{\ell}^*_{\boldsymbol{k}_2}(Y)$ from $\big(\delta_{\boldsymbol{k}_3}, \delta_{\boldsymbol{k}_3}^{d-(\mathsf{n}_\mathsf{x}-2)}(\hat{\Psi}_{\mu_\mathsf{x}}(\delta_{\boldsymbol{k}_3}; \boldsymbol{z}_x) + \alpha_{\boldsymbol{k}_2}\hat{\Phi}_{\mu_\mathsf{x}}(\delta_{\boldsymbol{k}_3}; \chi_{k_1})) \cdot \hat{v}(\delta_{\boldsymbol{k}_3}) - \alpha_{\boldsymbol{k}_2}\delta_{\boldsymbol{k}_3}\bar{p}_{k_1}(\delta_{\boldsymbol{k}_3})\hat{\Psi}_{\mu_\mathsf{x}}(\delta_{\boldsymbol{k}_3}; \boldsymbol{z}_y) - \delta_{\boldsymbol{k}_3}^{d+2}\hat{h}(\delta_{\boldsymbol{k}_3}) - \delta_{\boldsymbol{k}_3}^{d+1}\vartheta\big)_{k_3 \in \mathcal{T}}$. This is inspired by the first row in Eq. (23).
6. For any k_1 and *any fixed* k_2: $\hat{r}^*_{k_1}(X)$ from

$$\left(\delta_{\boldsymbol{k}_3}, \frac{\delta_{\boldsymbol{k}_3}^{d-(\mathsf{n}_\mathsf{x}-1)}\hat{f}(\delta_{\boldsymbol{k}_3}) - \bar{p}_{k_1}(\delta_{\boldsymbol{k}_3})}{\delta_{\boldsymbol{k}_3}^{d-(\mathsf{n}_\mathsf{x}-1)}(\delta_{\boldsymbol{k}_3}^{\mathsf{n}_\mathsf{x}} - \chi_{k_1})}\right)_{k_3 \in \mathcal{T}}.$$

(Since $\delta_{\boldsymbol{k}_3} \neq 0$ and $\delta_{\boldsymbol{k}_3}^{\mathsf{n}_\mathsf{x}} \neq \chi_{k_1}$, the denominator is non-zero.) This is inspired by the second row in Eq. (23).

Due to interpolation, all resulting polynomials $\hat{t}^* \in \mathcal{F} := \{\hat{f}, \hat{v}, \bar{p}_{k_1}, \hat{r}_{k_1}, \hat{h}_{\boldsymbol{k}_2}, \bar{\ell}_{\boldsymbol{k}_2}\}$ have degree $\leq d = n-1$. Since KZG is transcript-homomorphic, we can construct KZG evaluation arguments to show that $\hat{t}^*(\delta_{\boldsymbol{k}_3})$ is correct for each $\hat{t}^* \in \mathcal{F}$.

Up to now, everything worked at the "polynomial IOP" level and did not involve polynomial commitments, except that we assumed KZG is transcript-homomorphic to be able to compute KZG evaluation arguments. Next, we rule out attack vectors that involve breaking the KZG.

- Let $\mathsf{bad}_{\mathsf{ib}}$ be the case that for some $\hat{t} \in \mathcal{F}$, $[\hat{t}(\tau)]_1 \neq [\hat{t}^*(\tau)]_1$ (for example, $[\hat{r}_{k_1}(\tau)]_1 \neq [\hat{r}^*_{k_1}(\tau)]_1$ for some k_1). If $\mathsf{bad}_{\mathsf{ib}} = \mathsf{true}$, then $\mathcal{B}_{\mathsf{ib}}$ breaks interpolation binding while the extractor aborts. Thus, we can assume $\mathsf{bad}_{\mathsf{ib}} = \mathsf{false}$.
- The polynomials were interpolated for an arbitrary subset $\mathcal{T}$ and (sometimes) for any fixed choice of k_2 or $\boldsymbol{k}_2$. Let $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}}$ be the event that some $\hat{t}^* \in \mathcal{F}$ does not interpolate the corresponding data (e.g., $(\delta_{\boldsymbol{k}_3}, \hat{f}(\delta_{\boldsymbol{k}_3}))_{k_3 \in \mathcal{T}}$) for *some* value of k_2 or $\boldsymbol{k}_2$ on the whole interval $[\kappa_\delta]$. In particular, $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}}$ covers the case where the interpolated polynomial $\hat{t}^*$ depends on the choice of "any fixed" $\boldsymbol{k}_2/k_2$. Then, there exists a subset $\mathcal{T}' \subset [\kappa_\delta]$ of size $d+1$, such that the interpolation polynomial $\hat{t}'$ through those points is not equal to $\hat{t}^*$. If $[\hat{t}(\tau)]_1 \neq [\hat{t}'(\tau)]_1$, then we have broken interpolation binding. Otherwise, we break binding by returning $([\hat{t}(\tau)]_1, \hat{t}^*(X), \hat{t}'(X))$.
  If $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{true}$, then either $\mathcal{B}_{\mathsf{b}}$ can break KZG's d-binding or $\mathcal{B}_{\mathsf{ib}}$ can break KZG's d-interpolation binding. $\mathcal{E}_{\mathsf{css}}$ aborts when $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{true}$.

Next, assume $\mathsf{bad}_{\mathsf{ib}} = \mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{false}$. Since $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{false}$, polynomials $\hat{t}^*$ for $\hat{t} \in \mathcal{F}$ have degree $\leq d$. Since $\kappa_\delta \geq d+1 > \max_{\hat{t} \in \mathcal{F}}(\deg(\hat{t}^*))$, $\hat{t}^*$ are unique polynomials that evaluate given values at all $\delta_{\boldsymbol{k}_3}$. Since $\mathsf{bad}_{\mathsf{ib}} = \mathsf{false}$, all $\hat{t}$ agree with the commitments. Thus, $\mathcal{E}_{\mathsf{css}}$ has extracted small-degree polynomials that agree with the commitments and openings. Given this, we simplify the notation by dropping the asterisks and unnecessary subindices, writing, e.g., $\hat{f}$ instead of $\hat{f}^*$ and $\hat{r}_{k_1}$ instead of $\hat{r}^*_{k_1}$. The proof proceeds as an AGM proof would, given the polynomials returned by the algebraic adversary as an explanation.

Handling $\hat{r}_{k_1}(Y)$. From Item 6, we know that the polynomial

$$F_{k_1}(Y) := Y^{d-(\mathsf{n}_\mathsf{x}-1)} \cdot (\hat{f}(Y) - (Y^{\mathsf{n}_\mathsf{x}} - \chi_{k_1})\hat{r}_{k_1}(Y)) - \bar{p}_{k_1}(Y)$$

has

$$\kappa_\delta = 2d+3 > 2d+1$$

distinct roots $\delta_{\boldsymbol{k}_3}$. Since $\deg(F_{k_1}) = \max((d-(\mathsf{n}_\mathsf{x}-1))+d, (d-(\mathsf{n}_\mathsf{x}-1))+\mathsf{n}_\mathsf{x}+d, d) \le 2d+1$, $F_{k_1}(Y) \equiv 0$ as a polynomial. Moreover, since $\hat{f}(Y) - (Y^{\mathsf{n}_\mathsf{x}} - \chi_{k_1})\hat{r}_{k_1}(Y)$ is a polynomial, then

$$\hat{p}_{k_1}(Y) := Y^{-d+(\mathsf{n}_\mathsf{x}-1)}\bar{p}_{k_1}(Y) \tag{24}$$

must also be a polynomial. Since $\deg(\bar{p}_{k_1}) \le d$, it means

$$\hat{f}(Y) = (Y^{\mathsf{n}_\mathsf{x}} - \chi_{k_1})\hat{r}_{k_1}(Y) + \hat{p}_{k_1}(Y) \text{ for } \deg(\hat{p}_{k_1}) \le \mathsf{n}_\mathsf{x} - 1 \ . \tag{25}$$

Handling $\hat{\ell}_{\boldsymbol{k}_2}(X)$. Recall that by Eq. (24), $\bar{p}_{k_1}(X) = X^{d-(\mathsf{n}_\mathsf{x}-1)}\hat{p}_{k_1}(X)$ (we switch from variable Y to X since $\bar{p}$ has to use the same variable as other polynomials). Next, consider the polynomial

$$\begin{aligned} G_{\boldsymbol{k}_2}(X) := {} & X^{d-(\mathsf{n}_\mathsf{x}-2)}\hat{v}(X)\left(\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_x) + \alpha_{\boldsymbol{k}_2}\hat{\Phi}_{\mu_\mathsf{x}}(X;\chi_{k_1})\right) \\ & - \alpha_{\boldsymbol{k}_2}X^{d-(\mathsf{n}_\mathsf{x}-2)}\hat{p}_{k_1}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_y) \\ & - \left(\hat{h}_{\boldsymbol{k}_2}(X)X^{d+2} + \vartheta X^{d+1} + \bar{\ell}_{\boldsymbol{k}_2}(X)\right) \ . \end{aligned}$$

Here, say, $\deg(\hat{h}_{\boldsymbol{k}_2}), \deg(\bar{\ell}_{\boldsymbol{k}_2}) \le d$. From Item 5, we know that $G_{\boldsymbol{k}_2}(X)$ has

$$\kappa_\delta \ge 2d+3 > 2d+2$$

distinct roots. Since $\deg(G_{\boldsymbol{k}_2}) \le \max((d-(\mathsf{n}_\mathsf{x}-2))+d+(\mathsf{n}_\mathsf{x}-1), (d-(\mathsf{n}_\mathsf{x}-2))+d+(\mathsf{n}_\mathsf{x}-1), d+1+(\mathsf{n}_\mathsf{x}-1), d+(d+2), d+1, d) = 2d+2$, $G_{\boldsymbol{k}_2}(X) \equiv 0$ as a polynomial. Since all other terms of $G_{\boldsymbol{k}_2}(X)$ divide by $X^{d-(\mathsf{n}_\mathsf{x}-2)}$, so does $\bar{\ell}_{\boldsymbol{k}_2}(X)$. Thus,

$$\hat{\ell}_{\boldsymbol{k}_2}(X) = X^{-d+(\mathsf{n}_\mathsf{x}-2)}\bar{\ell}_{\boldsymbol{k}_2}(X)$$

is a polynomial of degree $\le \mathsf{n}_\mathsf{x} - 2$. Thus, from $G_{\boldsymbol{k}_2}(X) \equiv 0$, we get

$$\begin{aligned} G'_{\boldsymbol{k}_2}(X) := {} & \hat{v}(X)\left(\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_x) + \alpha_{\boldsymbol{k}_2}\hat{\Phi}_{\mu_\mathsf{x}}(X;\chi_{k_1})\right) - \alpha_{\boldsymbol{k}_2}\hat{p}_{k_1}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X;\boldsymbol{z}_y) \\ & - \left(\hat{h}_{\boldsymbol{k}_2}(X)X^{\mathsf{n}_\mathsf{x}} + \vartheta X^{\mathsf{n}_\mathsf{x}-1} + \hat{\ell}_{\boldsymbol{k}_2}(X)\right) \equiv 0 \ . \end{aligned}$$

For any $k_1 \in [\kappa_\chi]$, let $\boldsymbol{A}_{k_1} := \mathsf{Van}(\alpha_{k_1 1}, \alpha_{k_1 2})$ be a Vandermonde matrix. (Recall $\kappa_\alpha = 2$.) By the definition of a transcript tree $\alpha_{k_1 1} \neq \alpha_{k_1 2}$ for any

$k_1 \in [\kappa_\chi]$. Thus, $\det \boldsymbol{A}_{k_1} \neq 0$. Recall $\boldsymbol{k}_2 = (k_1, k_2)$. Now, $G'_{\boldsymbol{k}_2}(X) \equiv 0$ holds for $k_2 \in [\kappa_\alpha]$. Multiplying the resulting equation system $\begin{pmatrix} G'_{k_1 1}(X) \\ G'_{k_1 2}(X) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ by $\boldsymbol{A}_{k_1}^{-1}$,

$$\begin{aligned} &\begin{pmatrix} \hat{v}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X; \boldsymbol{z}_x) - \vartheta X^{\mathsf{n}_\mathsf{x}-1} \\ \hat{v}(X)\hat{\Phi}_{\mu_\mathsf{x}}(X; \chi_{k_1}) - \hat{p}_{k_1}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X; \boldsymbol{z}_y) \end{pmatrix} \\ &= \boldsymbol{A}_{k_1}^{-1} \begin{pmatrix} \hat{h}_{k_1 1}(X)X^{\mathsf{n}_\mathsf{x}} + \hat{\ell}_{k_1 1}(X) \\ \hat{h}_{k_1 2}(X)X^{\mathsf{n}_\mathsf{x}} + \hat{\ell}_{k_1 2}(X) \end{pmatrix} =: \begin{pmatrix} \hat{h}'_{k_1 1}(X)X^{\mathsf{n}_\mathsf{x}} + \hat{\ell}'_{k_1 1}(X) \\ \hat{h}'_{k_1 2}(X)X^{\mathsf{n}_\mathsf{x}} + \hat{\ell}'_{k_1 2}(X) \end{pmatrix} \end{aligned} \tag{26}$$

for some polynomials $\hat{h}'_{\boldsymbol{k}_2}(X)$ and $\hat{\ell}'_{\boldsymbol{k}_2}(X)$ where $\hat{\ell}'_{\boldsymbol{k}_2}(X)$ satisfy

$$\deg(\hat{\ell}'_{\boldsymbol{k}_2}) \leq \mathsf{n}_\mathsf{x} - 2 \ . \tag{27}$$

Next, by Eq. (25), $\deg(\hat{p}_{k_1}) \leq \mathsf{n}_\mathsf{x} - 1$. However, we only know $\deg(\hat{v}) \leq d$. Denote $\hat{V} := \mathsf{mod}_{\mathsf{n}_\mathsf{x}}(\hat{v})$. The coefficient of $X^{\mathsf{n}_\mathsf{x}-1}$ in $\hat{v}(X)\hat{\Phi}_{\mu_\mathsf{x}}(X; \chi_{k_1})$ is $\hat{V}(\chi_{k_1})$ by a simple generalization of Eq. (18). The coefficient of $X^{\mathsf{n}_\mathsf{x}-1}$ in $\hat{p}_{k_1}(X)\hat{\Psi}_{\mu_\mathsf{x}}(X; \boldsymbol{z}_y)$ is $\tilde{p}_{k_1}(\boldsymbol{z}_y)$ by Eq. (9). Since the right-hand side of the second row of Eq. (26) is $\hat{h}'_{k_1 2}(X)X^{\mathsf{n}_\mathsf{x}} + \hat{\ell}'_{k_1 2}(X)$ with $\deg(\hat{\ell}'_{k_1 2}) \leq \mathsf{n}_\mathsf{x} - 2$ by Eq. (27), its coefficient of $X^{\mathsf{n}_\mathsf{x}-1}$ is 0. Thus, the coefficient of $X^{\mathsf{n}_\mathsf{x}-1}$ in the left-hand side of the second row of Eq. (26) is 0, and hence $\hat{V}(\chi_{k_1}) = \tilde{p}_{k_1}(\boldsymbol{z}_y)$.

Similarly, the first row of Eq. (26) together with Lemma 2 gives $\tilde{V}(\boldsymbol{z}_x) = \vartheta$. Hence, Eqs. (26) and (27) give us

$$\tilde{V}(\boldsymbol{z}_x) = \vartheta \text{ and } \hat{V}(\chi_{k_1}) = \tilde{p}_{k_1}(\boldsymbol{z}_y) \ . \tag{28}$$

Using Lemma 1. Eq. (25) guarantees that the conditions of Lemma 1 are satisfied (when using $\hat{f}(Y)$ instead of $\hat{f}^*(Y)$, $\hat{p}_{k_1}(Y)$ instead of $\hat{p}^*(Y)$, and χ_{k_1} instead of x). Writing $\hat{f}(Y) = \sum_{i=0}^{\mathsf{n}_\mathsf{x}-1} f_i(Y^{\mathsf{n}_\mathsf{x}})Y^i$, the bivariate polynomial $Q(X, Y) = \sum_{i=0}^{\mathsf{n}_\mathsf{x}-1} f_i(X)Y^i$ satisfies

$$\hat{f}(Y) = Q(Y^{\mathsf{n}_\mathsf{x}}, Y) \ , \tag{29}$$

$$Q(\chi_{k_1}, Y) = \hat{p}_{k_1}(Y) \ . \tag{30}$$

Using Lemma 3. We have established all conditions of Lemma 3 (when using $\hat{V}$ instead of $\hat{v}$):

(1) $\hat{f}(Y) = Q(Y^{\mathsf{n}_\mathsf{x}}, Y)$ (see Eq. (29)),
(2) for $\kappa_\chi = \mathsf{n}_\mathsf{x}$ distinct values of $\chi_{k_1} \in \mathbb{F}$, $\hat{p}_{k_1}(Y) := Q(\chi_{k_1}, Y)$ (see Eq. (30)) has degree $\leq \mathsf{n}_\mathsf{x} - 1 = \mathsf{n}_\mathsf{y} - 1$ (see Eq. (25)) and satisfies $\tilde{p}_{k_1}(\boldsymbol{z}_y) = \hat{V}(\chi_{k_1})$ (see Eq. (28)).
(3) $\tilde{V}(\boldsymbol{z}_x) = \vartheta$ (see Eq. (28)).

Thus, by Lemma 3, $\tilde{f}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \vartheta$.

Thus, $\mathcal{E}_{\mathsf{css}}$ succeeds when $\mathsf{bad}_{\mathsf{b}\vee\mathsf{ib}} = \mathsf{bad}_{\mathsf{ib}} = \mathsf{false}$. More precisely,

$$\mathsf{Adv}^{\mathrm{ss}}_{\mathsf{PGen},\mathrm{Dolphin},\boldsymbol{\kappa},\mathcal{E}_{\mathsf{css}},\mathcal{A}_{\mathsf{css}}}(\lambda) \leq \mathsf{Adv}^{\mathrm{ib}}_{\mathsf{PGen},\mathsf{PC},d,\mathcal{B}_{\mathsf{ib}}}(\lambda) + \mathsf{Adv}^{\mathrm{b}}_{\mathsf{PGen},\mathsf{PC},d,\mathcal{B}_{\mathsf{b}}}(\lambda) \ .$$

By the assumption, the right-hand side is negligible. This proves the claim.

The field-size lower bound comes from the fact that δ is sampled from $\mathbb{F}^* \setminus \{t : t^{\mathsf{n}_\times} = \chi\}$, which means that the challenge space must have at least κ_δ elements. □

On the AGM security of Dolphin$_{agm}$. The computational special-soundness proof consists of two parts. In the first part, we extract from the field-element part of the transcript tree several polynomials, such as $\hat{f}^*(Y)$. Assuming KZG is binding and interpolation binding, we prove that these polynomials are well-defined, have low degree, and are consistent with polynomial commitments in the tree. Because the tree is accepting, we can then show that certain intermediate polynomials are equal to zero, and from this derive (by using Lemmas 1 and 3) that the prover was honest.

Dolphin$_{agm}$ is obtained from Dolphin by omitting sanitization (ϱ will only be used for batching), as evident from Fig. 2. An AGM [FKL18] knowledge-soundness proof of Dolphin$_{agm}$ has a trivial first part (the extracted polynomials are provided by the algebraic adversary as an explanation), but adds a reduction to PDL when some intermediate polynomials are not zeros. More precisely, the verifier equation states that $\hat{L}(\tau) = 0$, for $\hat{L}(X)$ defined by Eq. (20) (but without the highlighted parts). An AGM extractor knows $\hat{L}(X)$. Thus, if $\hat{L}(X) \neq 0$, then one can break the PDL. Otherwise, after using Schwartz-Zippel on ϱ and δ, one obtains $F(Y) = G(X) = 0$, where $F(Y)$ and $G(X)$ are like $F_{k_1}(Y)$ and $G_{\boldsymbol{k}_2}(X)$ in the proof of Theorem 2, except they do not depend on $\boldsymbol{k}_2$. Similarly to the proof of Theorem 2, one can then derive that the prover is honest. An AGMOS [LPS23] proof is only slightly more complicated.

Acknowledgments. We thank Janno Siim for useful comments. This study was co-funded by the European Union and the Estonian Research Council via project TEM-TA119. This work was supported by the Estonian Research Council grant PRG2531.

References

AFK22. Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-Shamir transformation of multi-round interactive proofs. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 113–142. Springer, Cham, November 2022. doi:10.1007/978-3-031-22318-1_5. 1, 1, 1.1, 2.1, 3.1, 4.4

AJM+23. Ittai Abraham, Philipp Jovanovic, Mary Maller, Sarah Meiklejohn, and Gilad Stern. Bingo: Adaptivity and asynchrony in verifiable secret sharing and distributed key generation. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 39–70. Springer, Cham, August 2023. doi:10.1007/978-3-031-38557-5_2. 1, 1.1, 1.1, 2.1, 6

BCC+16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien

Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_12. 4.1

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. doi:10.1007/978-3-031-07085-3_15. 1

BDFG20. Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Efficient polynomial commitment schemes for multiple points and polynomials. Cryptology ePrint Archive, Report 2020/081, 2020. URL: https://eprint.iacr.org/2020/081. 7

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. doi:10.1007/978-3-030-45721-1_24. 1, 2.1

BGH19. Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Report 2019/1021, 2019. URL: https://eprint.iacr.org/2019/1021. 4.1

BGM17. Sean Bowe, Ariel Gabizon, and Ian Miers. Scalable multi-party computation for zk-SNARK parameters in the random beacon model. Cryptology ePrint Archive, Report 2017/1050, 2017. URL: https://eprint.iacr.org/2017/1050. 1, 1, 4.3

Bro01. Daniel R. L. Brown. The exact security of ECDSA. Contributions to IEEE P1363a, January 2001. http://grouper.ieee.org/groups/1363/. 1

CDLT25. Rutchathon Chairattana-Apirom, Nico Döttling, Anna Lysyanskaya, and Stefano Tessaro. Everlasting Anonymous Rate-Limited Tokens, December 8–12, 2025. doi:10.1007/978-981-95-5119-4_14.

CDS94. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Berlin, Heidelberg, August 1994. doi:10.1007/3-540-48658-5_19. 1, 4, 3

CFF+21. Matteo Campanelli, Antonio Faonio, Dario Fiore, Anaïs Querol, and Hadrián Rodríguez. Lunar: A toolbox for more efficient universal and updatable zkSNARKs and commit-and-prove extensions. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 3–33. Springer, Cham, December 2021. doi:10.1007/978-3-030-92078-4_1. 1

CGG+24. Arka Rai Choudhuri, Sanjam Garg, Aarushi Goel, Sruthi Sekar, and Rohit Sinha. SublonK: Sublinear prover PlonK. *PoPETs*, 2024(3):314–335, July 2024. doi:10.56553/popets-2024-0080. 1, 1, 1.1, 3

CGKY25. Alessandro Chiesa, Ziyi Guan, Christian Knabenhans, and Zihan Yu. On the Fiat-Shamir Security of Succinct Arguments from Functional Commitments, May 21, 2025. Last accessed version from 2025.10.27. URL: https://eprint.iacr.org/2025/902. 1.1, 1.1, 2.1

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. doi:10.1007/978-3-030-45721-1_26. 1, 1, 3

DL08. Giovanni Di Crescenzo and Helger Lipmaa. Succinct NP Proofs from an Extractability Assumption. In Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe, editors, *Computability in Europe, CIE 2008*, volume 5028 of *LNCS*, pages 175–185, Athens, Greece, June 15–20, 2008. Springer, Heidelberg. 1

DP25. Benjamin E. Diamond and Jim Posen. Succinct arguments over towers of binary fields. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 93–122. Springer, Cham, May 2025. doi:10.1007/978-3-031-91134-7_4. 1.1, 4.2

EG25. Liam Eagen and Ariel Gabizon. MERCURY: A multilinear polynomial commitment scheme with constant proof size and no prover FFTs. Cryptology ePrint Archive, Report 2025/385, 2025. URL: https://eprint.iacr.org/2025/385. 1, 1, 1, 1, 2, 4.1, 4.1, 4.2, 4.3, 4.4

FFR24. Antonio Faonio, Dario Fiore, and Luigi Russo. Real-world universal zk-SNARKs are non-malleable. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 3138–3151. ACM Press, October 2024. doi:10.1145/3658644.3690351. 1

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Cham, August 2018. doi:10.1007/978-3-319-96881-0_2. 1, 4.4

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12. 1, 1.1, 4.3

Gab. Ariel Gabizon. Private communication. 1

GPS25. Chaya Ganesh, Sikhar Patranabis, and Nitin Singh. Samaritan: Linear-time prover SNARK from new multilinear polynomial commitments. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part V*, volume 16249 of *LNCS*, pages 456–489. Springer, Singapore, December 2025. doi:10.1007/978-981-95-5116-3_15. 1, 1, 1, 1, 1.1, 1.1, 2, 3, 3, 4.1, 4.1, 4.2, 4.3, 4.4

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Berlin, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_19. 1

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. doi:10.1007/978-3-662-49896-5_11. 1, 1.1

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. URL: https://eprint.iacr.org/2019/953. 1, 3

Ica09. Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 303–316. Springer, Berlin, Heidelberg, August 2009. doi:10.1007/978-3-642-03356-8_18. 1

JM24. Joseph Jaeger and Deep Inder Mohan. Generic and algebraic computation models: When AGM proofs transfer to the GGM. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part V*, volume

14924 of *LNCS*, pages 14–45. Springer, Cham, August 2024. doi:10.1007/978-3-031-68388-6_2. 1

KLPS26. Erki Külaots, Helger Lipmaa, Roberto Parisella, and Janno Siim. Special Soundness and Binding Properties: A Framework for Tightly Secure zk-SNARKs. In Emmanuel Thomé and Joan Daemen, editors, *EUROCRYPT 2026*, volume ? of *LNCS*, Rome, Italy, May 10–14, 2026. Springer, Cham. Accepted. 1.1, 1.1, 1.1, 1.1, 3

KT23. Tohru Kohrita and Patrick Towa. Zeromorph: Zero-knowledge multilinear-evaluation proofs from homomorphic univariate commitments. Cryptology ePrint Archive, Report 2023/917, 2023. URL: https://eprint.iacr.org/2023/917. 1

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Berlin, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_11. 1, 2.1

LFKN90. Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *31st FOCS*, pages 2–10. IEEE Computer Society Press, October 1990. doi:10.1109/FSCS.1990.89518. 1

Lip12. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Berlin, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_10. 1

LPS23. Helger Lipmaa, Roberto Parisella, and Janno Siim. Algebraic group model with oblivious sampling. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part IV*, volume 14372 of *LNCS*, pages 363–392. Springer, Cham, November / December 2023. doi:10.1007/978-3-031-48624-1_14. 1, 4.4

LPS24. Helger Lipmaa, Roberto Parisella, and Janno Siim. Constant-size zk-SNARKs in ROM from falsifiable assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 34–64. Springer, Cham, May 2024. doi:10.1007/978-3-031-58751-1_2. 1, 1.1, 1.1, 2, 2.1, 4.3, 4.4

LPS25. Helger Lipmaa, Roberto Parisella, and Janno Siim. On knowledge-soundness of plonk in ROM from falsifiable assumptions. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 362–395. Springer, Cham, August 2025. doi:10.1007/978-3-032-01907-3_12. 1, 1, 1.1, 1.1, 2.1, 3, 3, 4.2, 4.3

LSZ22. Helger Lipmaa, Janno Siim, and Michal Zajac. Counting vampires: From univariate sumcheck to updatable ZK-SNARK. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 249–278. Springer, Cham, December 2022. doi:10.1007/978-3-031-22966-4_9. 1

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. doi:10.1145/3319535.3339817. 4.1

Mic94. Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994. doi:10.1109/SFCS.1994.365746. 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. doi:10.1007/978-3-031-68403-6_8. 2.1

PST13. Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of correct computation. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 222–242. Springer, Berlin, Heidelberg, March 2013. doi:10.1007/978-3-642-36594-2_13. 1

RZ21. Carla Ràfols and Arantxa Zapico. An algebraic framework for universal and updatable SNARKs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 774–804, Virtual Event, August 2021. Springer, Cham. doi:10.1007/978-3-030-84242-0_27. 1

SPS25. Nitin Singh, Sikhar Patranabis, and Sayani Sinha. HybridPlonk: SubLogarithmic Linear Time SNARKs from Improved Sum-Check, May 21, 2025. Last checked version from August 29, 2025. URL: https://eprint.iacr.org/2025/908. 1

ST25. Srinath Setty and Justin Thaler. Twist and shout: Faster memory checking arguments via one-hot addressing and increments. Cryptology ePrint Archive, Report 2025/105, 2025. URL: https://eprint.iacr.org/2025/105. 1.1, 4.2

ZBK+22. Arantxa Zapico, Vitalik Buterin, Dmitry Khovratovich, Mary Maller, Anca Nitulescu, and Mark Simkin. Caulk: Lookup arguments in sublinear time. Cryptology ePrint Archive, Report 2022/621, 2022. URL: https://eprint.iacr.org/2022/621. 1, 1, 1, 1.1, 3

Zha22. Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Cham, August 2022. doi:10.1007/978-3-031-15982-4_3. 1