

Mist: Transparent Polynomial Commitments from Class Group Subset-Product Accumulators

Sriram Sridhar

University of California, Berkeley

Abstract

We construct the first polynomial commitment in groups of unknown order that simultaneously requires no trusted setup, has quasilinear prover time, and has constant-size opening proofs. Our main technical contribution is a transparent class group realization of the subset-product accumulator relation introduced by Morphic accumulators (CRYPTO '26). It supports accumulating a fixed family of class group elements and proving prescribed algebraic relations among the accumulated elements, yielding constant-size range arguments with quasilinear prover time.

Combining these range proofs with the DewTwo framework (CRYPTO '25) gives the first transparent polynomial commitment with quasilinear prover time and constant-size openings. Concretely, our construction also gives the shortest known opening proofs (1.3 KB) among transparent polynomial commitments. We prove security under falsifiable class group assumptions and also develop new extraction techniques over dyadic rationals, which may be of independent interest.

1 Introduction

1.1 Motivation and design goals

Succinct proof systems are protocols between a prover and a verifier that allow the prover to convince the verifier that it has performed some computation correctly without the verifier having to perform the computation itself. Since their inception [GMR85], there has been a great deal of work on them [Kil92, GKR08, GGPR13, Gro16, CHM⁺20, MBKM19, ACFY25] and they have found applications in many areas including verifiable machine learning [LXZ21], zk virtual machines [Suc26, RIS26], cryptocurrencies [BCG⁺14], and more.

Polynomial commitments (PCS) are a fundamental building block of succinct proof systems. A PCS allows a prover to commit to a polynomial and later prove that the polynomial evaluates to a certain value at a certain point. There exist standard blueprints for building proof systems that compose polynomial IOPs with polynomial commitments [CHM⁺20] to construct succinct proof systems. Consequently, the efficiency of the polynomial commitment directly determines the efficiency of the proof system built from it. There are multiple properties that we would like a polynomial commitment to have, and we focus on the following:

- **Transparent setup.** A trusted setup like that in [KZG10] produces its parameters using secret material that must then be destroyed; whoever holds that secret can forge proofs. A transparent setup derives the parameters from public coins instead, leaving nothing that can leak.

- **Prover time.** Applications such as zkVMs [RIS26, Suc26] prove statements over extremely large witnesses, and prover overhead is already the bottleneck, so a polynomial commitment with minimal prover overhead is what decides the largest instance one can handle. While the best prover time we can hope for is linear in the degree of the polynomial, current linear-time polynomial commitments [GLS⁺23, NS24] have much larger proof sizes.
- **Proof size.** The proof is the object that is transmitted, and in blockchain settings stored and paid for on chain. As the statements being proved grow, a proof that grows with them, even logarithmically, can become too large. The ideal is a proof of constant size, independent of the statement.
- **Falsifiable assumptions.** We want the strongest security under the weakest assumptions. A proof in an idealized model such as the generic group model rules out only adversaries that respect the idealization, yet several of the constructions we compare against are proved secure only there. We would rather rest on assumptions that can be stated as a game and attacked directly.

1.2 Contributions

Class group SPAcc (subset-product accumulators) and range proofs. We give the first transparent realization of the subset-product relation [PTX26], using prime-leading representatives of class group elements and a succinct form-equivalence argument. Instantiating the accumulated family by $X_i = 2^{i-1} \cdot G$ yields batched integer range proofs over any number of intervals of arbitrary widths, with a quasilinear prover and a proof consisting of two group elements and 16 field elements regardless of how many intervals are batched.

Dyadic exponent arguments. Underpinning this, we show that DewTwo’s integer modular-consistency assumption is unusable in class groups, and in any group with efficient square roots, and replace it with *dyadic* modular consistency: falsifiable, implied by its integer counterpart, and sound in a generic group model that exposes the halving oracle class groups really have. Under it the PoKE and PoKEMath extractors run unchanged over $\mathbb{Z}[1/2]$, and adaptive-root and low-order security follow.

The Mist polynomial commitment. Replacing DewTwo’s recursive range layer with one batched SPAcc argument gives Mist, a polynomial commitment that is transparent, has a quasilinear prover, an $O(\log N)$ verifier, and constant-size openings, under dyadic modular consistency and discrete log in class groups. An opening is three class group elements and 14 field elements, or 1312 bytes at $\lambda = 128$ – the shortest proof size we know of among transparent polynomial commitments. Like DewTwo, this is the cost at an evaluation point sampled independently of the commitment, which is what the standard polynomial-IOP compilers ask for; arbitrary points cost one further repetition, which we sketch in Section 7.3.

1.3 Comparison with prior work

Constant-sized proofs or in general proofs independent of the size of the statement (in some form) are not common in the transparent setting. One family where we do see it is accumulators in groups of unknown order [BP97, CL02, BBF19, PTX26], which admit efficient membership and non-membership proofs. Another is proofs of exponentiation [Wes19, BBF19], whose size is independent of the bit length of the integer witness.

When considering constant-sized polynomial commitment schemes, all the constructions we know

Scheme	Trans.	Falsif.	$ \mathbf{pp} $	Prover	Verifier	$ \pi $	$(N = 2^{30})$
KZG [KZG10]	×	✓	$N \mathbb{G}_P$	$N \mathbb{G}_P$	1 pairing	$1 \mathbb{G}_P$	32 B
Bulletproofs [BBB ⁺ 18]	✓	✓	$N \mathbb{G}$	$N \mathbb{G}$	$N \mathbb{G}$	$2 \log N \mathbb{G}$	1.9 KB
WHIR [ACFY25]	✓	✓	$O(1)$	$N \log N \mathbb{F}$	$\lambda \log N \log \log N \mathbb{H}$	$\lambda \log N \log \log N \mathbb{H}$	105 KB
Dew [AGL ⁺ 23]	✓	×	$O(1) \mathbb{G}_U$	$N \mathbb{G}_U, N^2 M(N) \mathbf{b}$	$\log N \mathbb{F}, O(1) \mathbb{G}_U$	$O(1) \mathbb{G}_U$	12 KB
Dew++ [Fu25]	✓	×	$O(1) \mathbb{G}_U$	$N \mathbb{G}_U, N^2 M(N) \mathbf{b}$	$\log N \mathbb{F}, O(1) \mathbb{G}_U$	$O(1) \mathbb{G}_U$	2.7 KB
Behemoth [SB25]	✓	×	$N \mathbb{G}_U$	$N \mathbb{G}_U, N^2 M(N) \mathbf{b}$	$\log N \mathbb{F}, O(1) \mathbb{G}_U$	$O(1) \mathbb{G}_U$	9 KB
Morphic [PTX26]	×	×	$O(1) \mathbb{G}_U$	$N \mathbb{G}_U$	$\log N \mathbb{F}, O(1) \mathbb{G}_U$	$O(1) \mathbb{G}_U$	
DewTwo [BMSS25]	✓	✓	$\log N \mathbb{G}_U$	$\lambda N \mathbb{G}_U, \lambda N \log^2 N \mathbf{b}$	$\log N \mathbb{F}, \log N \mathbb{G}_U$	$\log \log N \mathbb{G}_U$	4.5 KB
Mist (this work)	✓	✓	$O(1) \mathbb{G}_U$	$\lambda N \mathbb{G}_U, \tilde{O}(\lambda N) \mathbf{b}$	$\log N \mathbb{F}, O(1) \mathbb{G}_U$	$O(1) \mathbb{G}_U$	1.3 KB

Table 1: Polynomial commitment schemes, with $O(\cdot)$ omitted for brevity. $\mathbb{G}_P$ is a pairing-group operation, $\mathbb{G}$ a discrete-log-group operation, $\mathbb{G}_U$ an operation in a group of unknown order, $\mathbb{F}$ a field operation, $\mathbb{H}$ a hash, and $\mathbf{b}$ a bit operation; $M(\cdot)$ is the cost of integer multiplication. Every scheme listed commits with a single group element. “Trans.” is a transparent setup and “Falsif.” a security proof under falsifiable assumptions rather than in the generic group model.

of use hidden-order groups [BFS20, AGL⁺23, Fu25, SB25, BMSS25, PTX26]. These all encode the polynomial as a single integer and differ mainly in how they discharge the resulting range checks, which is what separates their prover times and proof sizes. One notable point here is that Behemoth’s [SB25] techniques are different and can in principle also give a constant verifier when using the techniques in this work. The closest to our work is Morphic [PTX26], which introduces the subset-product relation we realize; their realization needs an RSA modulus, so the setup is trusted, and it is analysed in the generic group model.

Another related primitive is SNARKs over the integers [CHA24, BMSS25, GWV⁺25, SSPP26, AGK⁺26, CXNB26], which can potentially be used to construct integer range proofs as they can handle arbitrary NP relations. Most of these works are based on the technique of fingerprinting to reduce proving integer equations to equivalent relations modulo small primes, and then compress them further. However, none of these currently supply a constant-size range proof (except in a circular fashion), so they cannot be used as a black box for the range checks a polynomial commitment needs.

1.4 Organization

Section 2 gives a technical overview, followed by preliminaries in Section 3. Sections 4 and 5 develop the class group assumptions and dyadic extraction toolbox. Section 6 constructs SPAcc and range proofs, and Section 7 designs our new polynomial commitment scheme from the above tools.

2 Techniques

Our goal is a polynomial commitment scheme that is transparent, has a quasilinear prover, and has constant-size openings. While we know how to obtain any combination of two of these properties, no known scheme achieves all three. First, we review the current techniques and bottlenecks in building these schemes, then we give a high-level overview of our new techniques and how we overcome these bottlenecks.

Most of the polynomial commitment schemes that achieve constant proof size (or close to constant

proof size) use groups of unknown order (GUOs) – these support committing to and proving relations over integers. Here, the recurring idea is to compress properties of many numbers into properties of fewer, larger ones, then prove those with a fingerprinting-style argument that is succinct in the bit length of the witness but not in the number of witness elements. Although proving integer relations ends up being relatively cheap, all these works require proving that the integers lie in certain ranges, requiring integer range proofs. This is where most of these works differ in their techniques, and where the bottlenecks arise.

Sums of squares. Dew, Dew++ and Behemoth [AGL⁺23, Fu25, SB25] certify nonnegativity (and hence arbitrary ranges) by writing an integer as a sum of three/four squares (Legendre’s theorem/Lagrange’s theorem). The witness is only a constant number of integers, so proofs are constant size and the setup stays transparent, but the best known algorithms to produce this decomposition are cubic. Improvements to these algorithms or other algorithms for constant-sized positivity witnesses that are prover efficient seem to be hard to find.

Non constant-sized decompositions. DewTwo [BMSS25] decomposes the relevant integers into $O(\log \log N)$ squares instead which can be done with a quasilinear prover. Unfortunately, the proof is no longer constant and compressing it needs further inner product machinery.

Accumulators. Morphic accumulators [PTX26] reuse a primitive that already has constant-size proofs: accumulators in groups of unknown order prove membership succinctly and efficiently. The primitive they isolate is a subset-product accumulator, which for a fixed public family $\mathbf{X} = (X_1, \dots, X_n)$ of group elements proves that a target Y is the product of a subset of them. Taking $X_i = 2^{i-1} \cdot G$ makes a subset exactly a binary expansion, so this immediately yields a range proof for $[0, 2^n)$.

The catch is that a standard accumulator accumulates primes, to avoid divisibility issues, whereas the objects here are group elements. Morphic accumulators bridge the two levels by attaching to each group element a canonical integer representative, accumulating those, and showing division intractability. Both halves of that bridge are RSA-specific: an element of an RSA group already *is* an integer modulo the modulus, which is what makes a canonical representative available and easy to accumulate. Most other groups such as class groups offer no such thing¹, so transparency is lost. This leaves us with two roadblocks to overcome in order to get a transparent polynomial commitment with quasilinear prover time and constant-size openings:

Extending existing integer proof machinery to class groups. The PoKE-style arguments [BBF19] that underlie all of this extract an *integer* witness, and their soundness is established either in the generic group model or for groups in which square roots are hard. Neither covers class groups, and as we explain below the failure is not just an artifact of the proofs.

The accumulator has to be redesigned from scratch. Class group elements carry no canonical integer representative, so we need another way to accumulate group elements and prove aggregates of them in the exponent.

¹Class groups are what we will end up using in this work to solve the below issues, for reasons that will be evident later.

2.1 New assumptions and dyadic extraction

Recall the shape of a PoKE protocol for a commitment $U = u \cdot G$. The verifier sends a prime ℓ ; the prover replies with the residue $r = u \bmod \ell$ together with a group element Q carrying the quotient, and the verifier checks $U = \ell \cdot Q + r \cdot G$. Extraction collects answers for many primes: the residues pin down u modulo their product, and once that product exceeds the witness the integer is determined.

In a class group this check can be met with no integer in mind at all. The class number is odd, so halving is efficient, and for $U = \text{Half}(G)$ the answers $r = (\ell + 1)/2$ and $Q = -U$ are accepted for every odd ℓ , since $\ell \cdot (-U) + \frac{\ell+1}{2} \cdot G = U$. They are perfectly consistent, but with the dyadic rational $1/2$: the integer CRT lift of the residues is $(L + 1)/2$, which opens U only if $L \cdot G = 0$.

DewTwo [BMSS25] moved these protocols out of the generic group model by isolating a falsifiable assumption, modular consistency (MCA), which says precisely that consistent answers to many coprime challenges force the integer CRT lift to open the commitment. They use Jacobians over hyperelliptic curves, where square roots are unavailable and the issue never arises – but this breaks in class groups.

We therefore introduce *dyadic modular consistency* (DMC), which draws the same conclusion over $\mathbb{Z}[1/2]$ rather than $\mathbb{Z}$. DMC is also falsifiable, is implied by MCA, and also holds in a generic group model extended with a halving oracle to model class groups. Under it the PoKE extractor runs unchanged and returns an element of $\mathbb{Z}[1/2]$, and so does PoKEMath, its extension that additionally certifies a low-degree relation $f(\mathbf{u}) = 0$ among the committed values. This answers the first roadblock, at the cost of every downstream protocol having to deal with witnesses over $\mathbb{Z}[1/2]$.

2.2 Subset-product accumulators from class groups

This is our main contribution, and it starts from an observation about what a class group element looks like. An element is an equivalence class of binary quadratic forms (a, b, c) of a fixed discriminant $\Delta = b^2 - 4ac$, under the action of $\text{SL}_2(\mathbb{Z})$. We normally work with the reduced representative, but every class has infinitely many others, and in particular infinitely many whose leading coefficient is prime. Primes are well suited to an accumulator, since distinct primes do not collide in a product and we do not have to worry about division intractability. So we preprocess the fixed family $\mathbf{X} = (X_1, \dots, X_n)$ by choosing for each X_i a prime-leading representative $F_i = (p_i, \beta_i, \gamma_i)$ with the p_i distinct.

The second ingredient is that composition (the group operation in class groups) is simple when leading coefficients are coprime: the composite has leading coefficient $\prod_i a_i$, and its middle coefficient is recovered from the b_i by the Chinese remainder theorem. With the family fixed this has a convenient consequence. Solving $B \equiv \beta_i \pmod{2p_i}$ once, simultaneously for all i produces a single integer B that serves *every* subset: for any $I \subseteq [n]$ the form

$$F(P_I, B) = \left(P_I, B, \frac{B^2 - \Delta}{4P_I} \right), \quad P_I = \prod_{i \in I} p_i,$$

is properly equivalent to the composite of $(F_i)_{i \in I}$, and so represents $\sum_{i \in I} X_i$. One preprocessed pair (T, B) , with $T = \prod_i p_i$, therefore describes all 2^n subsets at once, and only the leading coefficient has to change from one subset to the next.

The protocol follows almost immediately. Preprocessing commits to (T, B) . To show that a target Y is the subset sum for I , the prover commits to $P = P_I$, to the cofactor $\bar{P} = T/P$, and to a matrix $M \in \text{SL}_2(\mathbb{Z})$ with $Y^M = F(P, B)$, and then proves in the exponent that $P\bar{P} = T$, that

$\det M = 1$, and that Y^M has leading coefficient P and middle coefficient B . These are low-degree relations among committed values, so one PoKEMath invocation discharges all of them, and the whole reduction transmits a single group element. Taking $X_i = 2^{i-1} \cdot G$ turns this into the range proof we were after, with the accumulator preprocessed once.

Two difficulties still remain. The first is the extractor returning dyadic rationals, so what it certifies is equivalence of Y and $F(P, B)$ under $\text{SL}_2(\mathbb{Z}[1/2])$, while the class group is defined by $\text{SL}_2(\mathbb{Z})$. A priori they could identify classes that ought to stay distinct. We rule this out by choosing the discriminant carefully: for $\Delta \equiv 5 \pmod{8}$ we can show that every integral form of that discriminant is anisotropic modulo 2, and a dyadic change of variables between two such forms is actually forced to be integral. The second is that preprocessing has to *find* the prime-leading representatives, which ends up being taking successive coprime pairs (r, t) and testing $X_i(r, t)$ for primality. Heuristically these values are no sparser in primes than random integers of the same size, so a few hundred trials suffice. This is still only one-time work, and easily updatable (extendable) to larger ranges.

2.3 Putting everything together

Now that we have a transparent range argument with constant-size proofs and a quasilinear prover, we can take the DewTwo polynomial commitment scheme, delete its recursive decomposition along with the inner-product machinery and drop our accumulator in its place.

The replacement simplifies the protocol as well, all checks that we need to make are now low-degree relations among committed values, so all of them collapse into one batched PoKEMath instance over four commitment blocks. The prover sends two commitments and a single quotient, and the verifier checks everything with one multi-scalar multiplication together with one evaluation of the joint predicate modulo the challenge prime.

Concretely, an opening is three class group elements and fourteen field elements. At $\lambda = 128$, with the 288-byte compressed class group representation of Section 4.1, that is 1312 bytes, independent of the degree. The prover is quasilinear in N , verifier work is $O(\log N)$, and security rests on DMC together with DLOG in class groups. To our knowledge this is the shortest opening among transparent polynomial commitments: the nearest schemes either send $O(\log \log N)$ group elements or pay a cubic prover for a constant-size one. Section 7 gives the construction in full.

3 Preliminaries

3.1 Notation and relations

We write $[n] := \{1, \dots, n\}$ and use bold lowercase letters for vectors. The i -th entry of $\mathbf{x}$ is denoted x_i , and $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_i x_i y_i$. For a finite set S , $s \leftarrow S$ denotes uniform sampling from S . All logarithms are base two. For an integer a , let $|a|_{\text{bit}} := \lceil \log(1 + |a|) \rceil$ denote its bit length. We write

$$\text{Primes}(\lambda) := \{\ell \in \mathbb{N} : \ell \text{ is prime and } |\ell|_{\text{bit}} = \lambda\}$$

for the set of primes of bit length λ . We write $M(n)$ for the cost of multiplying two n -bit integers and use $\tilde{O}(\cdot)$ to suppress factors polylogarithmic in the argument and polynomial terms in the security parameter. Our costs account separately for multi-scalar multiplications (as they are the most expensive operations in general), writing $\text{MSM}(m, b)$ for the cost of an m -term multi-scalar multiplication with b -bit scalars.

All groups considered in this paper are abelian and are written additively. For $a \in \mathbb{Z}$ and G in a group $\mathbb{G}$, $a \cdot G$ denotes scalar multiplication, and $\langle \mathbf{a}, \mathbf{G} \rangle := \sum_i a_i \cdot G_i$. Besides the integers $\mathbb{Z}$, we also use the ring of dyadic rationals

$$\mathbb{Z}[1/2] := \{a/2^k : a \in \mathbb{Z}, k \in \mathbb{N}\}.$$

For an odd prime ℓ , reduction modulo ℓ is well defined on this ring: $(a/2^k) \bmod \ell = a(2^k)^{-1} \in \mathbb{F}_\ell$. We define relations $\mathcal{R}$ as consisting of triples $(\mathbf{pp}, x, w)$, where $\mathbf{pp}$ contains public parameters, x is an instance, and w is a witness. For fixed public parameters, we write

$$\mathcal{R}^{\mathbf{pp}} := \{(x, w) : (\mathbf{pp}, x, w) \in \mathcal{R}\}.$$

The polynomial-time algorithm $\text{Setup}(1^\lambda, N)$ samples the public parameters. Here N bounds the supported instance size; we leave it implicit when it is clear from context.

3.1.1 Multilinear polynomials

We denote the space of μ -variate multilinear polynomials over a field $\mathbb{F}$ by $\mathbb{F}^{\leq 1}[X_1, \dots, X_\mu]$. Put $N := 2^\mu$, and let $\text{bin}(i, j)$ be the j -th bit (MSB first) of the μ -bit representation of $i \in \{0, \dots, N-1\}$. Every $p \in \mathbb{F}^{\leq 1}[X_1, \dots, X_\mu]$ has a unique coefficient representation

$$p(\mathbf{X}) = \sum_{i=0}^{N-1} p_i \prod_{j=1}^{\mu} X_j^{\text{bin}(i,j)}.$$

We identify p with its coefficient vector $\mathbf{p} = (p_0, \dots, p_{N-1}) \in \mathbb{F}^N$. For $\mathbf{z} \in \mathbb{F}^\mu$, define its multilinear monomial/evaluation vector $\chi(\mathbf{z}) \in \mathbb{F}^N$ by $\chi_i(\mathbf{z}) := \prod_j z_j^{\text{bin}(i,j)}$; then we have the evaluation $p(\mathbf{z}) = \langle \mathbf{p}, \chi(\mathbf{z}) \rangle$.

3.2 Reductions and arguments of knowledge

We recall reductions of knowledge from [KP23]. Consider two relations $\mathcal{R}_1$ and $\mathcal{R}_2$ with the same Setup algorithm. A *reduction of knowledge* from $\mathcal{R}_1$ to $\mathcal{R}_2$ is a tuple of probabilistic polynomial-time algorithms $\Pi = (\text{Setup}, \mathcal{P}, \mathcal{V})$, where $\text{Setup}(1^\lambda, N) \rightarrow \mathbf{pp}$ samples public parameters for instances of size at most N , and the interaction $\langle \mathcal{P}(\mathbf{pp}, x_1, w_1), \mathcal{V}(\mathbf{pp}, x_1) \rangle$ on a source instance-witness pair (x_1, w_1) has common output a target instance x_2 , with the prover additionally outputting a target witness w_2 . It must be *complete*, in that an honest interaction on $(\mathbf{pp}, x_1, w_1) \in \mathcal{R}_1$ outputs $(x_2, w_2) \in \mathcal{R}_2^{\mathbf{pp}}$ with probability one; *knowledge sound*, in that for every expected polynomial-time adversary producing an accepting interaction there is an expected polynomial-time extractor recovering w_1 with $(\mathbf{pp}, x_1, w_1) \in \mathcal{R}_1$ except with negligible probability; and *publicly reducible*, in that x_2 is a deterministic function of $(\mathbf{pp}, x_1)$ and the transcript. Appendix A.3 states the three properties in full, together with the sequential composition theorem of [KP23] that lets us chain reductions. We write $\Pi : \mathcal{R}_1 \rightarrow \mathcal{R}_2$.

An *argument of knowledge*² for $\mathcal{R}$ is the special case of a reduction of knowledge to the trivial relation $\Pi : \mathcal{R} \rightarrow \mathcal{R}_\top$, where $\mathcal{R}_\top := \{(\mathbf{pp}, \text{true}, \perp) : \mathbf{pp} \text{ is a valid public parameter}\}$. Unless stated otherwise, all arguments and reductions in this paper are public coin, interactive and adaptive; the Fiat–Shamir transform [FS87] is used when a non-interactive argument is needed. We use the term *succinct* qualitatively, for a reduction or argument whose communication and verification cost are

²We use the terms “proof” and “argument” interchangeably throughout this paper.

at most polylogarithmic in the size of the witness; the precise bounds obtained by each protocol are stated in the corresponding theorem statements.

3.3 Subset-product accumulators

Let $\mathbb{G}$ be a group and let $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{G}^n$ be a public family of group elements. We define the subset-product relation, in our additive notation, by

$$\mathcal{R}_{\text{SPAcc}} := \{((\mathbb{G}, \mathbf{X}), Y, \mathbf{b}) : \mathbf{X} \in \mathbb{G}^n, Y \in \mathbb{G}, \mathbf{b} \in \{0, 1\}^n, Y = \langle \mathbf{b}, \mathbf{X} \rangle\}.$$

Equivalently, the witness may be written as the subset $I := \{i : b_i = 1\}$ and the relation requires $Y = \sum_{i \in I} X_i$. A *subset-product accumulator* is a succinct argument of knowledge for $\mathcal{R}_{\text{SPAcc}}$. This relation abstracts the subset-product relation of [PTX26].

3.4 Integer range proofs

Fix vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^m$ such that $a_j \leq b_j$ for every $j \in [m]$, and let $\mathbf{G} = (G_1, \dots, G_m) \in \mathbb{G}^m$ be public bases. The batched integer range relation indexed by these endpoints is

$$\mathcal{R}_{\text{Range}}^{\mathbf{a}, \mathbf{b}} := \{((\mathbb{G}, \mathbf{G}), V, \mathbf{v}) : \mathbf{G} \in \mathbb{G}^m, V \in \mathbb{G}, \mathbf{v} \in \mathbb{Z}^m, V = \langle \mathbf{v}, \mathbf{G} \rangle, a_j \leq v_j \leq b_j \ (j \in [m])\}.$$

The single-value relation $\mathcal{R}_{\text{Range}}^{a, b}$ is the case $m = 1$. An *integer range proof* is a succinct argument of knowledge for this relation. Note that this is a stronger primitive than a range proof over a finite field, as we don't restrict the range to be a subset of the field.

3.5 Commitment schemes

A commitment scheme for a message space $\mathcal{M}$ consists of three probabilistic polynomial-time (PPT) algorithms $\text{CS} = (\text{Setup}, \text{Commit}, \text{Open})$ with the following syntax:

- $\text{Setup}(1^\lambda, N) \rightarrow \text{pp}_{\text{CS}}$ samples public parameters for messages of size at most N .
- $\text{Commit}(\text{pp}_{\text{CS}}, m) \rightarrow (C, d)$, for $m \in \mathcal{M}$, returns a commitment C and opening hint d .
- $\text{Open}(\text{pp}_{\text{CS}}, C, m, d) \rightarrow \{0, 1\}$ verifies an opening.

Correctness requires that for all $m \in \mathcal{M}$,

$$\Pr \left[\text{Open}(\text{pp}_{\text{CS}}, C, m, d) = 1 \mid \begin{array}{l} \text{pp}_{\text{CS}} \leftarrow \text{Setup}(1^\lambda, N), \\ (C, d) \leftarrow \text{Commit}(\text{pp}_{\text{CS}}, m) \end{array} \right] = 1.$$

The scheme is *binding* if for every PPT adversary $\mathcal{A}$,

$$\Pr \left[\begin{array}{l} m_0 \neq m_1 \wedge \\ \text{Open}(\text{pp}_{\text{CS}}, C, m_0, d_0) = 1 \wedge \\ \text{Open}(\text{pp}_{\text{CS}}, C, m_1, d_1) = 1 \end{array} \mid \begin{array}{l} \text{pp}_{\text{CS}} \leftarrow \text{Setup}(1^\lambda, N); \\ (C, m_0, d_0, m_1, d_1) \leftarrow \mathcal{A}(\text{pp}_{\text{CS}}) \end{array} \right] \leq \text{negl}(\lambda).$$

3.6 Polynomial commitment schemes

A multilinear polynomial commitment scheme over $\mathbb{F}_q$ is a tuple

$$\text{PCS} = (\text{Setup}, \text{Commit}, \text{Open}, \text{Eval})$$

with the following syntax, writing $\mathbb{F}_q^{\leq 1}[\mathbf{X}]$ for $\mathbb{F}_q^{\leq 1}[X_1, \dots, X_\mu]$:

- $\text{Setup}(1^\lambda, N) \rightarrow \text{pp}_{\text{PCS}}$ samples public parameters for polynomials in $\mathbb{F}_q^{\leq 1}[\mathbf{X}]$ of size at most N .
- $\text{Commit}(\text{pp}_{\text{PCS}}, p) \rightarrow (C, d)$, for $p \in \mathbb{F}_q^{\leq 1}[\mathbf{X}]$, returns a commitment C and opening hint d .
- $\text{Open}(\text{pp}_{\text{PCS}}, C, p, d) \rightarrow \{0, 1\}$ verifies an opening.
- Eval is a succinct argument of knowledge (Section 3.2) for the evaluation relation $\mathcal{R}_{\text{Eval}}$ defined below.

We require $\text{CS} := (\text{Setup}, \text{Commit}, \text{Open})$ to be a binding commitment scheme (Section 3.5) with message space $\mathbb{F}_q^{\leq 1}[X_1, \dots, X_\mu]$. Evaluation at $\mathbf{z} \in \mathbb{F}_q^\mu$ is captured by the ternary relation

$$\mathcal{R}_{\text{Eval}} := \left\{ (\text{pp}_{\text{PCS}}, (q, \mu, C, \mathbf{z}, y), (p, d)) : \begin{array}{l} \mu \in \mathbb{N}, \mathbf{z} \in \mathbb{F}_q^\mu, y \in \mathbb{F}_q, p \in \mathbb{F}_q^{\leq 1}[X_1, \dots, X_\mu], \\ \text{Open}(\text{pp}_{\text{PCS}}, C, p, d) = 1, p(\mathbf{z}) = y \end{array} \right\}.$$

This property also implies the weaker notion of evaluation binding.

We use two orders of quantification for evaluation knowledge soundness, following [BMSS25]. In the *adaptive* notion, the adversary chooses $(C, \mathbf{z}, y)$ after seeing pp . In the *weak* notion, the adversary first fixes C , the point $\mathbf{z}$ is then sampled from the prescribed public distribution independently of that commitment, and only afterward may the adversary choose y and run the evaluation argument. In both cases the extractor must return (p, d) satisfying $\mathcal{R}_{\text{Eval}}$. We state explicitly which notion is achieved by each construction.

4 Class Groups and Assumptions

4.1 Groups of unknown order

A *group of unknown order* (GUO) is a finite abelian group in which group operations and sampling can be performed efficiently, but whose order is infeasible to compute. Although exponents are mathematically reduced modulo the group order, no known polynomial-time algorithm can compute that reduced representative. This and related hard problems makes GUOs useful for integer commitments [DF02], accumulators [BBF19], verifiable delay functions [Wes19], and arguments about integer arithmetic [CHA24, GWV⁺25].

The main GUO candidates that we know of are RSA groups [RSA78], ideal class groups of imaginary quadratic orders [BW88], and Jacobians of genus-3 hyperelliptic curves [Lee20]. RSA groups require either a trusted modulus generator or a distributed modulus-generation protocol [BF97, CHI⁺21] while class groups and hyperelliptic Jacobians admit transparent generation. DewTwo chooses genus-3 Jacobians for their short representations [BMSS25, DGS20] to reduce proof size as well as to avoid extraction issues due to efficient square root computations [BS96] being possible in class groups. We choose class groups despite these issues: in addition to transparent setup, every ideal class is represented by infinitely many primes, which is the feature used by our SPAcc construction and is hence non-black box in the GUO.

Following [DGS20], we use the parameter tuple $(\lambda, k) = (128, 40)$ and a 3072-bit discriminant. The class number is approximately 2^{1536} and, except with probability about 2^{-40} , the best known generic attack costs about 2^{128} operations. A class group element occupies about 384 bytes without compression and 288 bytes with the compression of [DGS20], versus about 192 bytes for the corresponding genus-3 Jacobian representation.

As in DewTwo, every PoKE-style challenge is sampled from $\text{Primes}(2\lambda)$. The extra factor of two accounts for subexponential attacks against the GUO hardness assumptions; λ -bit challenges do

not in general provide λ bits of soundness in this parameter regime [BMSS25, DGS20].

4.2 Class groups as binary quadratic forms

Let $\Delta < 0$ be a nonsquare integer with $\Delta \equiv 1 \pmod{4}$, and let

$$\mathcal{O}_\Delta := \mathbb{Z} \left[\frac{\Delta + \sqrt{\Delta}}{2} \right].$$

We use the quadratic-form presentation of the class group $\text{Cl}(\mathcal{O}_\Delta)$.³ Though the ideal class group presentation is more common and may be easier for proofs, this representation is more computation-friendly (corresponding to actual class group arithmetic implementations [BCIL23]) and we also exploit properties of this representation in our construction. A form is a primitive positive-definite triple

$$f = (a, b, c), \quad a > 0, \quad \gcd(a, b, c) = 1, \quad b^2 - 4ac = \Delta.$$

For $M = \begin{pmatrix} r & s \\ t & u \end{pmatrix} \in \text{SL}_2(\mathbb{Z})$, define the change of variables $f^M(X, Y) := f(rX + sY, tX + uY)$:

$$f^M = (ar^2 + brt + ct^2, 2ars + b(ru + st) + 2ctu, as^2 + bsu + cu^2). \quad (1)$$

Proper equivalence is the orbit relation under this action. A form is *reduced* when

$$-a < b \leq a \leq c, \quad b \geq 0 \text{ if } a = c.$$

Every proper-equivalence class contains exactly one reduced form [BV07].

Reduction is by the classical normalize-and-swap loop, which also returns the transforming matrix; we give it as Algorithm A.1 in Appendix A.2.

Write $\text{red}(f)$ for the quadratic form returned by $\text{Reduce}(f)$. Gauss composition followed by reduction is the class group law (optimized algorithms like NUCOMP [Coh93] interleave the two for efficiency). We only explicitly present the following special case when the leading coefficients are coprime – the general case is not needed for the constructions in this paper.

Algorithm 4.1: ComposeCoprime($f_1, \dots, f_m$)

Input: Forms $f_i = (a_i, b_i, c_i)$ of discriminant Δ with odd leading coefficients such that $\gcd(a_i, a_j) = 1$ for $i \neq j$.

Output: The reduced representative of $[f_1] + \dots + [f_m]$.

Set $A \leftarrow \prod_i a_i$ and solve

$$B \equiv b_i \pmod{2a_i} \quad (i \in [m])$$

by the generalized Chinese remainder theorem. Since $B^2 \equiv \Delta \pmod{4A}$, return

$$\text{red} \left(A, B, \frac{B^2 - \Delta}{4A} \right).$$

For the discriminants $\Delta \equiv 1 \pmod{4}$ used here,

$$0 = \text{red} \left(1, 1, \frac{1 - \Delta}{4} \right), \quad -(a, b, c) = \text{red}(a, -b, c).$$

³The ideal presentation sends (a, b, c) to the class of $a\mathbb{Z} + ((-b + \sqrt{\Delta})/2)\mathbb{Z}$.

If $p \nmid \Delta$ is an odd prime and $b^2 \equiv \Delta \pmod{4p}$, then

$$F_{p,b} := \left(p, b, \frac{b^2 - \Delta}{4p} \right) \quad (2)$$

is a *prime form*. The roots b and $-b$ give us additive inverses. Every class contains infinitely many prime forms, so we may choose distinct pairwise-coprime leading coefficients for **ComposeCoprime**.⁴ Every protocol in this paper draws its public parameters from the same transparent procedure, and from nothing else: a group description together with however many independent bases the protocol needs.

Algorithm 4.2: $\text{Setup}_{\text{CG}}(1^\lambda, k)$

Output: A class group description $\mathbf{gk}$ and k independent bases.

1. Derive a 3072-bit prime $p_\Delta \equiv 3 \pmod{8}$ from public coins and set $\Delta := -p_\Delta \equiv 5 \pmod{8}$. Put $\mathbf{gk} := (\Delta, \mathcal{C}_\Delta)$, where $\mathcal{C}_\Delta = \text{Cl}(\mathcal{O}_\Delta)$ is represented by reduced forms with the operations above.
2. Derive $G_1, \dots, G_k \leftarrow \mathcal{C}_\Delta$ from public coins using a hash-to-class-group map, and return $(\mathbf{gk}, G_1, \dots, G_k)$.

We write $\text{CGGen}(1^\lambda)$ for step 1 alone, which is what the assumptions below are stated over. Step 2 needs care: a naive hash algorithm may not be distribution-correct [SBK24]. Since both steps consume only public coins, the setup is transparent, and the number k of bases is the only thing that varies between protocols – nine for the accumulator of Section 6, and a constant number more for the range and evaluation arguments. The consequence of $\Delta \equiv 5 \pmod{8}$ is used in Section 6 to obtain rigidity of integral forms under dyadic changes of variables.

4.3 Assumptions

Let

$$\mathbf{gk} \leftarrow \text{CGGen}(1^\lambda), \quad \mathbb{G} := \text{Cl}(\mathcal{O}_\Delta), \quad \mathbf{G} \leftarrow \mathbb{G}^n.$$

For our prime discriminants the class number is odd, so doubling is bijective; moreover, because the factorization of Δ is public, this unique half can be computed efficiently [TCLM21]. Write $\text{Half}(G)$ for the unique $H \in \mathbb{G}$ such that $2 \cdot H = G$, and let Half^e denote its e -fold iteration, with $\text{Half}^0(G) = G$. For $a/2^e \in \mathbb{Z}[1/2]$, define $(a/2^e) \cdot G := a \cdot \text{Half}^e(G)$. While our honest prover does not need to use this operation, an adversary may use it, and we need to reason about it while proving knowledge soundness.

For pairwise coprime $\ell = (\ell_1, \dots, \ell_s)$, write $L := \prod_i \ell_i$ and let $\text{CRT}((r_i)_i, (\ell_i)_i) \in [0, L)$ denote the unique integer congruent to r_i modulo ℓ_i for every i . We also use its centered variant $\text{CRT}^\pm((r_i)_i, (\ell_i)_i) \in (-L/2, L/2]$, the unique such representative in this shifted range.

Definition 4.1 (Discrete logarithm [BBF19]). For polynomial n , the discrete-log assumption holds if

$$\text{Adv}_{\mathcal{A}, n}^{\text{DLOG}} := \Pr[\mathbf{x} \neq \mathbf{y} \wedge \langle \mathbf{x}, \mathbf{G} \rangle = \langle \mathbf{y}, \mathbf{G} \rangle] \leq \text{negl}(\lambda)$$

for every PPT $\mathcal{A}$, where $\mathbf{gk} \leftarrow \text{CGGen}(1^\lambda)$, $\mathbf{G} \leftarrow \mathbb{G}^n$, and $(\mathbf{x}, \mathbf{y}) \leftarrow \mathcal{A}(\mathbf{gk}, \mathbf{G})$ with $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$.

⁴The efficient search for these representatives and its cost are discussed in Section 6.

DewTwo [BMSS25] moved PoKE-style extraction out of the generic group model by isolating a falsifiable *modular consistency* assumption (MCA): informally, an adversary that fixes U in advance and then answers many pairwise-coprime challenges ℓ_i consistently must be answering about an integer, in the sense that the coordinate-wise integer CRT lift of its responses opens U . As Appendix B.1 shows, this is unusable in class groups, where efficient halving lets a prover answer every challenge consistently while the integer lift opens nothing. We therefore generalize it to admit dyadic lifts.

We generalize the above assumption to a dyadic variant that is sound in class groups and fulfills the same purpose as MCA – allowing for extraction of a witness from a prover that can answer many challenges for PoKE-like protocols.

For $d \in \mathbb{N}$ and vector residues $(\mathbf{r}_i)_{i=1}^s \in (\mathbb{Z}^n)^s$, applying CRT and $\text{CRT}^\pm$ coordinate-wise, define the candidate set

$$\mathcal{X}_{d,\ell}((\mathbf{r}_i)_i) := \left\{ \text{CRT}((r_{i,j})_i, (\ell_i)_{i \in [n]}) \right\} \cup \left\{ 2^{-e} \cdot \text{CRT}^\pm((2^e r_{i,j})_i, (\ell_i)_{i \in [n]}) : 0 \leq e \leq d \right\}. \quad (3)$$

This is the set of denominator-depth- d dyadic lifts of $(\mathbf{r}_i)_i$; it has at most $d + 2$ elements and is computable in polynomial time.

Definition 4.2 (Dyadic modular consistency). Let $B_T := 2^{T+2}$ and $d_T := T$. DMC holds if, for all polynomial n, T, s , every $(T \cdot \text{rep})$ -step $\mathcal{A}_1$, and every PPT $\mathcal{A}_2$,

$$\text{Adv}_{\mathcal{A}, T, n, s}^{\text{DMC}} := \Pr \left[\begin{array}{l} \forall i \in [s] : U = \langle \mathbf{r}_i, \mathbf{G} \rangle + \ell_i \cdot Q_i \\ \wedge \forall \mathbf{x} \in \mathcal{X} : U \neq \langle \mathbf{x}, \mathbf{G} \rangle \end{array} \mid \begin{array}{l} \mathbf{gk} \leftarrow \text{CGGen}(1^\lambda); \\ \mathbf{G} \leftarrow \mathbb{G}^n; \\ (U, \text{st}) \leftarrow \mathcal{A}_1(\mathbf{gk}, \mathbf{G}); \\ [(\mathbf{r}_i, Q_i, \ell_i)]_{i=1}^s \leftarrow \mathcal{A}_2(\text{st}); \\ \ell_i \in \text{Primes}(2\lambda) \text{ distinct}; \\ L := \prod_i \ell_i > 2B_T 2^{d_T}; \\ \mathcal{X} \leftarrow \mathcal{X}_{d_T, \ell}((\mathbf{r}_i)_i) \end{array} \right] \leq \text{negl}(\lambda).$$

Appendix A.1 gives evidence for DMC by proving it in a generic group model augmented with the halving operation available in our class group interface.

Formally, $\text{MCA}_{T, n, s}$ is Definition 4.2 with $\mathcal{X}$ replaced by the singleton integer lift $\{\text{CRT}((r_{i,j})_i, (\ell_i)_{i \in [n]})\}$, the challenges required only to be pairwise coprime, and no lower bound on L ; as there, $\mathcal{A}_1$ has $(T \cdot \text{rep})$ steps and $\mathcal{A}_2$ returns s transcripts, and DewTwo states the diagonal case $s = T$. The game is written out as Definition A.2 in Appendix A.4. Since the integer lift always lies in $\mathcal{X}_{d,\ell}$, MCA implies DMC.

We show the relation to existing assumptions (Adaptive Root, Low Order) in Figure 1.

$$\begin{aligned} \text{AR} : \quad & (U, \text{st}) \leftarrow \mathcal{A}_1(\mathbf{gk}), \quad \ell \leftarrow \text{Primes}(2\lambda), \\ & X \leftarrow \mathcal{A}_2(\text{st}, \ell) : \quad \ell \cdot X = U \neq 0; \\ \text{LO} : \quad & (W, m) \leftarrow \mathcal{A}(\mathbf{gk}) : \quad m \cdot W = 0, \quad W \neq 0, \quad m \neq 0. \end{aligned}$$

Each assumption says that the corresponding event has negligible probability.

The corresponding lemma statements, proofs, and the GGM evidence for these assumptions appear in Appendix A.

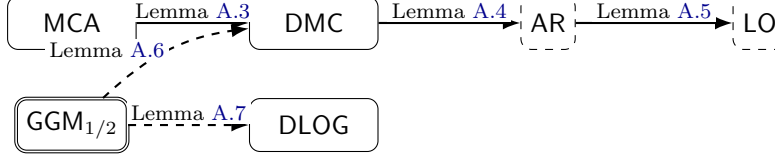


Figure 1: Relations among the assumptions and derived properties. Solid arrows are reductions; dashed arrows are evidence in the root-aware generic group model of Appendix A.1.

5 Dyadic Extraction for PoKE-style Protocols

This section develops the extraction techniques and presents the subprotocols used throughout the rest of the paper. We present the PoKE protocol [BBF19] first, explain why efficient square roots obstruct integer extraction in class groups, and then give the batched dyadic PoKEMath argument to which the later protocols reduce their algebraic relations. These are proved knowledge sound under the time-bounded dyadic modular-consistency (DMC) assumption defined in Section 4; both proofs appear in Appendix C.

5.1 Dyadic PoKE

The proof of knowledge of exponent protocol (PoKE) [BBF19] lets a prover demonstrate knowledge of the exponents in a multi-base commitment. Its public parameters are $\mathbf{pp} := (\mathbf{gk}, \mathbf{G})$ as produced by Algorithm 4.2, so that in particular the bases are sampled by the setup rather than chosen by the prover; the verifier sends a prime $\ell \leftarrow \text{Primes}(2\lambda)$, the prover returns the canonical residues $\mathbf{r} = \mathbf{u} \bmod \ell$ together with a group element Q carrying the quotient, and the verifier accepts if every r_j lies in $[0, \ell)$ and $U = \ell \cdot Q + \langle \mathbf{r}, \mathbf{G} \rangle$.

As sketched in Section 2, integer extraction fails in any group with efficient halving: for $U := \text{Half}(G)$ the responses $Q := -U$ and $r := (\ell + 1)/2$ are accepted on every odd challenge, and the integer CRT lift of those residues opens U only in the low-order event. The transcripts encode $1/2$, so we work instead with the dyadic relation

$$\mathcal{R}_{\text{PoKE}}^{\mathbb{Z}[1/2]} := \{((\mathbf{gk}, \mathbf{G}), U, \mathbf{u}) : \mathbf{u} \in \mathbb{Z}[1/2]^m \text{ and } U = \langle \mathbf{u}, \mathbf{G} \rangle\}, \quad (4)$$

for which the protocol runs unchanged – for odd ℓ every $u_j \in \mathbb{Z}[1/2]$ still has a canonical residue $r_j \in [0, \ell)$ – and is an argument of knowledge under DMC. The protocol and security proof are given in Appendix B.

5.2 Batched dyadic PoKEMath

PoKEMath [BMSS25] augments PoKE with an arithmetic predicate on the committed exponents: the verifier additionally checks that the responses satisfy the predicate modulo the challenge. The reductions used in this work apply one predicate to several commitments at once, so we state only this batched form; the single-commitment protocol is the case $k = 1$.

Fix $k \geq 1$ and block dimensions $m_1, \dots, m_k$, and put $m := \sum_j m_j$. The public parameters are $\mathbf{pp} := (\mathbf{gk}, (\mathbf{G}_j)_{j \in [k]})$ with each $\mathbf{G}_j \in \mathbb{G}^{m_j}$ sampled independently by Algorithm 4.2. For $f \in \mathbb{Z}[(X_{j,h})_{j,h}]$, evaluated naturally over $\mathbb{Z}[1/2]$, define

$$\mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2],k} := \left\{ ((\mathbf{gk}, (\mathbf{G}_j)_{j \in [k]}), ((U_j)_{j \in [k]}, f), (\mathbf{u}_j)_{j \in [k]}) \mid \right. \\ \left. \mathbf{u}_j \in \mathbb{Z}[1/2]^{m_j}, \quad U_j = \langle \mathbf{u}_j, \mathbf{G}_j \rangle \ (j \in [k]), \quad f(\mathbf{u}_1, \dots, \mathbf{u}_k) = 0 \right\}. \quad (5)$$

The same prime challenge is shared by every block, and a second challenge compresses all quotient elements into one. However, the response vectors must be fixed before this batching challenge.

Protocol 5.1: BatchPoKEMath

Common input: A class group description $\mathbf{gk}$, independently sampled base vectors $\mathbf{G}_j$, commitments $U_j \in \mathbb{G}$ for $j \in [k]$, and a joint polynomial f .

Prover input: Vectors $\mathbf{u}_j \in \mathbb{Z}[1/2]^{m_j}$ satisfying Equation (5).

1. The verifier samples one $\ell \leftarrow \text{Primes}(2\lambda)$ and sends ℓ .
2. For every $j \in [k]$, the prover computes

$$\mathbf{r}_j := \mathbf{u}_j \bmod \ell, \quad \mathbf{q}_j := \frac{\mathbf{u}_j - \mathbf{r}_j}{\ell},$$

and sends only $(\mathbf{r}_j)_{j \in [k]}$.

3. After the responses are fixed, the verifier samples $\eta \leftarrow \mathbb{F}_\ell$ and sends η . Both parties set ω_j to the canonical integer representative of $\eta^{j-1} \in \mathbb{F}_\ell$.
4. The prover sends the single quotient element

$$Q := \sum_{j=1}^k \omega_j \langle \mathbf{q}_j, \mathbf{G}_j \rangle.$$

The verifier accepts if

$$\sum_{j=1}^k \omega_j U_j = \ell Q + \sum_{j=1}^k \omega_j \langle \mathbf{r}_j, \mathbf{G}_j \rangle, \quad (6)$$

$$f(\mathbf{r}_1, \dots, \mathbf{r}_k) = 0 \pmod{\ell}, \quad (7)$$

and every response lies in $[0, \ell)$.

Theorem 5.1 (Batched PoKEMath). *Under DMC and DLOG, Protocol 5.1 is a public-coin argument of knowledge for $\mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2],k}$ for polynomial k . Suppose the honest prover's witness lies in $\mathbb{Z}^m$ and each coordinate has bit length at most L_w , put*

$$L_q := \max\{L_w, 2\lambda\} + 1,$$

and let C_f be the number of $\mathbb{F}_\ell$ operations needed to evaluate f . Then:

- Prover: $\text{MSM}(m, L_q + 2\lambda)$ group operations and $O(mM(L_q))$ bit operations.
- Verifier: one $(k + m + 1)$ -term MSM whose scalars have at most $4\lambda + 1$ bits, and $O((C_f + m)M(2\lambda))$ bit operations.
- Proof size: one group element and m elements of $\mathbb{F}_\ell$.

The extractor recovers an individual quotient element for each block from several batching challenges by Vandermonde inversion, applies the dyadic PoKE extractor blockwise, and then turns Equation (7) into an equality over $\mathbb{Z}[1/2]$ using the adaptive-root property implied by DMC. The proof appears in Appendix C.

Remark 5.2 (Challenge ordering is essential). The responses must precede η . For example, if ℓ and a nonzero η were known first for two openings $C_1 = a_1 G$ and $C_2 = a_2 G$, a prover could choose any

nonzero $\delta \in \mathbb{F}_\ell$ and set

$$r_1 = a_1 + \delta, \quad r_2 = a_2 - \eta^{-1}\delta \pmod{\ell}.$$

The aggregate congruence would hold although neither response need be the correct individual residue. Fixing (r_1, r_2) before sampling η prevents this cancellation and enables the Vandermonde extraction above.

6 Subset-Product Accumulators and Range Proofs

6.1 SPAcc from class groups

We now give a class group construction of the relation from Section 3.3. For a fixed family of reduced forms $\mathbf{X} = (X_1, \dots, X_n) \in \mathcal{C}_\Delta^n$, write the relation

$$\mathcal{R}_{\text{SPAcc}}^{\text{gk}, \mathbf{X}} := \left\{ (Y, I) : I \subseteq [n] \text{ and } Y = \sum_{i \in I} X_i \right\}. \quad (8)$$

6.1.1 Construction

Preprocessing. For every $i \in [n]$, find a potentially non-reduced prime-leading representative⁵

$$F_i = (p_i, \beta_i, \gamma_i), \quad \text{red}(F_i) = X_i,$$

where the p_i are distinct odd primes not dividing Δ . Each β_i is odd, since $\beta_i^2 \equiv \Delta \pmod{4p_i}$ with Δ odd, so the congruences below agree modulo 2 and are simultaneously solvable even though the moduli $2p_i$ share that factor. Running the composition algorithm from Algorithm 4.1 on $(F_i)_{i \in [n]}$, up to but not including its final call to **red**, determines via the generalized CRT

$$T := \prod_{i=1}^n p_i, \quad B \in [0, 2T), \quad B \equiv \beta_i \pmod{2p_i} \quad (i \in [n]), \quad (9)$$

uniquely modulo $2T$, with $B^2 \equiv \Delta \pmod{4T}$. The same congruences hold for any subfamily, so for every $I \subseteq [n]$ and $P_I := \prod_{i \in I} p_i$, the integral form

$$F(P_I, B) := \left(P_I, B, \frac{B^2 - \Delta}{4P_I} \right) \quad (10)$$

is properly equivalent to the output of **ComposeCoprime** $((F_i)_{i \in I})$ before reduction, and so represents $\sum_{i \in I} X_i$. The two are not equal: that output carries the middle coefficient $B_I \in [0, 2P_I)$ determined by the subfamily alone, and $B \equiv B_I \pmod{2P_I}$, so $\begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}$ with $n = (B - B_I)/2P_I$ carries one to the other. Using the global B for every subset is what lets one preprocessed pair (T, B) serve all of them, and the slack is absorbed by the variable κ below.

Derive independent bases

$$A_T, A_B, \mathbf{H} = (H_P, H_{\bar{P}}, H_r, H_s, H_t, H_u, H_\kappa) \leftarrow \mathcal{C}_\Delta^9$$

from public coins, and define the fixed commitment

$$C_{\text{fix}} := T \cdot A_T + B \cdot A_B. \quad (11)$$

⁵Section 6.1.2 gives the search algorithm and its cost.

Preprocessing outputs a prover key and verification key

$$\text{pk}_{\text{SPA}_{\text{Acc}}} := (\mathbf{X}, (F_i)_i, T, B) \text{vk}_{\text{SPA}_{\text{Acc}}} := (\text{gk}, A_T, A_B, \mathbf{H}, C_{\text{fix}}),$$

The online verifier using the succinct key never needs to read the prime forms or the raw integers T and B . There can be a one-time validation step, if needed.

For a target $Y = (a, b, c)$, define polynomials in variables $\mathbf{Z} = (\tau, \beta, P, \bar{P}, r, s, t, u, \kappa)$

$$\begin{aligned} h_{\text{prod}}(\mathbf{Z}) &:= P\bar{P} - \tau, \\ h_{\text{det}}(\mathbf{Z}) &:= ru - st - 1, \\ h_{\text{lead}}(\mathbf{Z}) &:= P - ar^2 - brt - ct^2, \\ h_{\text{mid}}(\mathbf{Z}) &:= \beta + 2\kappa P - 2ars - b(ru + st) - 2ctu, \end{aligned}$$

and set

$$\Phi_Y(\mathbf{Z}) := h_{\text{prod}}(\mathbf{Z})^2 + h_{\text{det}}(\mathbf{Z})^2 + h_{\text{lead}}(\mathbf{Z})^2 + h_{\text{mid}}(\mathbf{Z})^2. \quad (12)$$

Since $\mathbb{Z}[1/2] \subset \mathbb{R}$, the equation $\Phi_Y = 0$ is equivalent to the simultaneous vanishing of all four constituent polynomials.

The target is the two-block form of the dyadic PoKEMath relation of Section 5.2, whose blocks are the fixed accumulator and online intermediate commitments:

$$\mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 2} = \left\{ \begin{array}{l} x = (\text{gk}, A_T, A_B, \mathbf{H}, \\ \quad C_{\text{fix}}, U, \Phi_Y), \\ w = (\tau, \beta, \mathbf{w}) \end{array} \left| \begin{array}{l} C_{\text{fix}} = \tau \cdot A_T + \beta \cdot A_B, \\ U = \langle \mathbf{w}, \mathbf{H} \rangle, \\ \Phi_Y(\tau, \beta, \mathbf{w}) = 0 \end{array} \right. \right\}. \quad (13)$$

Protocol 6.1: Reduction $\Pi_{\text{SPA}_{\text{Acc}}}$

Preprocessing input: A class group description gk and a fixed family $\mathbf{X} = (X_1, \dots, X_n) \in \mathcal{C}_{\Delta}^n$.

1. For every $i \in [n]$, run [Algorithm B.1](#) to obtain a prime-leading form $F_i = (p_i, \beta_i, \gamma_i)$ with distinct odd prime $p_i \nmid \Delta$ and $\text{red}(F_i) = X_i$. Compute T and B as in Equation (9).
2. Derive $A_T, A_B, \mathbf{H}$ from public coins, compute C_{fix} as in Equation (11), and run the validation checks above. Output

$$\begin{aligned} \text{pk}_{\text{SPA}_{\text{Acc}}} &= (\mathbf{X}, (F_i)_i, T, B), \\ \text{vk}_{\text{SPA}_{\text{Acc}}} &= (\text{gk}, A_T, A_B, \mathbf{H}, C_{\text{fix}}). \end{aligned}$$

Common input: The verification key $\text{vk}_{\text{SPA}_{\text{Acc}}}$ and a reduced form $Y = (a, b, c) \in \mathcal{C}_{\Delta}$.

Prover input: The prover key $\text{pk}_{\text{SPA}_{\text{Acc}}}$ and a subset $I \subseteq [n]$ such that $Y = \sum_{i \in I} X_i$.

1. The prover sets

$$P := \prod_{i \in I} p_i, \quad \bar{P} := \prod_{i \notin I} p_i,$$

and computes $M = \begin{pmatrix} r & s \\ t & u \end{pmatrix} \in \text{SL}_2(\mathbb{Z})$ such that $Y^M = F(P, B)$, and sets $\kappa := 0$. More generally, the prover may use any $M \in \text{SL}_2(\mathbb{Z})$ and $\kappa \in \mathbb{Z}$ satisfying

$$P = ar^2 + brt + ct^2, \quad (14)$$

$$B + 2\kappa P = 2ars + b(ru + st) + 2ctu. \quad (15)$$

It sets $\mathbf{w} := (P, \bar{P}, r, s, t, u, \kappa)$, forms

$$U := \langle \mathbf{w}, \mathbf{H} \rangle,$$

and sends U to the verifier.

2. The parties output (for $\mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 2}$)

$$\begin{aligned} \mathcal{P} &\rightarrow ((\mathbf{gk}, A_T, A_B, \mathbf{H}, C_{\text{fix}}, U, \Phi_Y), (T, B, \mathbf{w})), \\ \mathcal{V} &\rightarrow (\mathbf{gk}, A_T, A_B, \mathbf{H}, C_{\text{fix}}, U, \Phi_Y). \end{aligned}$$

The reduction transmits only U . In the composed argument, PoKEMath sends one quotient group element for the two commitments in Equation (13).

6.1.2 Dyadic rigidity of integral forms

The extractor for the preceding protocol recovers elements of $\mathbb{Z}[1/2]$, even though an honest prover uses only integers. The following property of the chosen discriminant prevents this relaxation from identifying different integral form classes. The proofs are deferred to Appendix C.

Lemma 6.1 (Dyadic rigidity of integral forms). *Let $\Delta < 0$ satisfy $\Delta \equiv 5 \pmod{8}$, and let f and g be primitive positive-definite integral forms of discriminant Δ . If $g = f^M$ for some $M \in \text{SL}_2(\mathbb{Z}[1/2])$, then $M \in \text{SL}_2(\mathbb{Z})$. Hence f and g are properly equivalent integral forms.*

Theorem 6.2 (Class-group SPAcc reduction). *Assuming DLOG, for every fixed family $\mathbf{X} \in \mathcal{C}_\Delta^n$, Protocol 6.1 is a reduction of knowledge*

$$\mathcal{R}_{\text{SPAcc}}^{\mathbf{gk}, \mathbf{X}} \longrightarrow \mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 2}.$$

Let L bound the bit length of Δ , T , B , the coefficients of Y , and every coordinate of the target witness $(T, B, \mathbf{w})$, and put $D := |\Delta|_{\text{bit}}$. The reduction costs are:

- Preprocessing: n searches for prime representatives (Section 6.1.2), n validation checks, $\text{MSM}(2, L)$ group operations, and $O(M(L) \log n)$ bit operations.
- Prover: $\text{MSM}(7, L)$ group operations and $O(M(L)(\log n + \log L))$ bit operations.
- Verifier: $O(D)$ bit operations.
- Communication: one group element.

Corollary 6.3 (SPAcc argument). *Assuming DMC and DLOG, the sequential composition of Π_{SPAcc} with the two-block form of PoKEMath is a public-coin argument of knowledge for $\mathcal{R}_{\text{SPAcc}}^{\mathbf{gk}, \mathbf{X}}$.*

Finding prime-leading representatives. Preprocessing needs, for every class X_i , a properly equivalent form whose leading coefficient is a distinct odd prime not dividing Δ . This is the most expensive part of the construction, though it is one-time work that depends only on $\mathbf{X}$ and can be extended if instance sizes grow. The leading coefficient of X^M is $X(r, t)$, where $(r, t)^\top$ is the first column of M , and any coprime pair (r, t) extends by Bézout to an element of $\text{SL}_2(\mathbb{Z})$; so it suffices to enumerate coprime pairs in order of increasing $\max\{|r|, |t|\}$ and test $X(r, t)$ for primality, and $\Delta \equiv 5 \pmod{8}$ makes every such value odd automatically. Reduction bounds the prime found by $O(|\Delta| \max\{|r|, |t|\}^2)$, which fixes $\rho(\lambda) = |\Delta|_{\text{bit}} + O(\log |\Delta|_{\text{bit}})$ as the bit length of a preprocessed prime – the quantity the costs above and in Section 6.3 are stated in terms of. Heuristically the values $X(r, t)$ contain primes at the density predicted for integers of their size, so $O(|\Delta|_{\text{bit}})$ candidates

suffice per index and the expected cost is $O(|\Delta|_{\text{bit}}^2 M(|\Delta|_{\text{bit}}))$ bit operations. Appendix B gives the search algorithm, its analysis, and an experiment supporting the density heuristic.

6.2 Batched SPAcc

One preprocessed family proves any number of subset-product statements at once, since the preprocessing depends only on $\mathbf{X}$ and not on the online targets. For $d \geq 1$, define the product relation

$$\left(\mathcal{R}_{\text{SPAcc}}^{\text{gk}, \mathbf{X}}\right)^d := \left\{ ((Y_h)_{h \in [d]}, (I_h)_{h \in [d]}) : I_h \subseteq [n], \quad Y_h = \sum_{i \in I_h} X_i \quad (h \in [d]) \right\}. \quad (16)$$

Use one copy of $(\mathbf{X}, (F_i)_i, T, B)$ and $C_{\text{fix}} = T \cdot A_T + B \cdot A_B$. For each target, derive an independent seven-base vector $\mathbf{H}_h$ and let $\mathbf{w}_h$ denote the seven coordinates used in Φ_{Y_h} . Define the joint predicate

$$\Phi_{\mathbf{Y}}^{(d)}(\tau, \beta, \mathbf{w}_1, \dots, \mathbf{w}_d) := \sum_{h=1}^d \Phi_{Y_h}(\tau, \beta, \mathbf{w}_h). \quad (17)$$

The range argument of Section 6.3 is the case $d = 2$: an interval check is two subset-product statements against the same family, one per endpoint. Ranges of *different* widths are batched not by raising d but by enlarging $\mathbf{X}$, each coordinate contributing its own block of powers, as in Corollary 6.5.

The reduction $\Pi_{\text{MultiSPAcc}}^{(d)}$ is then Protocol 6.1 with three changes, and we defer the full description to Appendix B. Preprocessing computes $(F_i)_i, T, B$ and C_{fix} exactly once and additionally derives d independent seven-base vectors $\mathbf{H}_1, \dots, \mathbf{H}_d$. The prover builds each witness $\mathbf{w}_h$ as before and sends the single online commitment $U := \sum_h \langle \mathbf{w}_h, \mathbf{H}_h \rangle$. The output is again a two-block PoKEMath instance, on (C_{fix}, U) with opening blocks (T, B) and $(\mathbf{w}_1, \dots, \mathbf{w}_d)$, now carrying the joint predicate $\Phi_{\mathbf{Y}}^{(d)}$.

Theorem 6.4 (Batched SPAcc).

Assume the discrete-log assumption of Definition 4.1. For every family $\mathbf{X} \in \mathcal{C}_{\Delta}^n$ and polynomial d , Protocol B.2 is a reduction of knowledge

$$\left(\mathcal{R}_{\text{SPAcc}}^{\text{gk}, \mathbf{X}}\right)^d \longrightarrow \mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 2}.$$

It performs the accumulator preprocessing once, sends one online group element, and produces PoKEMath blocks of dimensions 2 and $7d$. Under DMC and the discrete-log assumption, composition with Protocol 5.1 sends two group elements and $2+7d$ field elements. Its proof appears in Appendix C.

6.3 Batched range proofs from SPAcc

Fix endpoint vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^m$ with $a_j \leq b_j$, put

$$n_j := \lceil \log(b_j - a_j + 1) \rceil, \quad n := \sum_{j \in [m]} n_j,$$

and sample independent public bases $\mathbf{G} = (G_1, \dots, G_m) \in \mathcal{C}_{\Delta}^m$. Preprocess the single concatenated family

$$\mathbf{X} := (2^{i-1} \cdot G_j : j \in [m], i \in [n_j]). \quad (18)$$

Its subsets are in bijection with the vectors $\mathbf{z}$ satisfying $0 \leq z_j < 2^{n_j}$: the block of $\mathbf{X}$ belonging to G_j carries the binary expansion of z_j , and the blocks have independent bases. One accumulator therefore covers m intervals of different widths, with only the block lengths n_j varying.

For an instance $V = \langle \mathbf{v}, \mathbf{G} \rangle$ put

$$Y_- := V - \langle \mathbf{a}, \mathbf{G} \rangle, \quad Y_+ := \langle \mathbf{b}, \mathbf{G} \rangle - V,$$

so that membership of every v_j in its interval is the single pair of subset-product statements (Y_-, Y_+) over $\mathbf{X}$. This is the $d = 2$ case of [Protocol B.2](#): both targets use the same family, the same prime representatives and the same (T, B) , and only the two seven-coordinate online witnesses are separate. With independent witness bases $\mathbf{H}_-, \mathbf{H}_+$ the joint predicate is

$$\Psi_{V, \mathbf{a}, \mathbf{b}}(\tau, \beta, \mathbf{w}_-, \mathbf{w}_+) := \Phi_{Y_-}(\tau, \beta, \mathbf{w}_-) + \Phi_{Y_+}(\tau, \beta, \mathbf{w}_+). \quad (19)$$

Protocol 6.2: Argument Range_{CG}

Preprocessing input: A class group description gk , independently sampled bases $\mathbf{G} \in \mathcal{C}_{\Delta}^m$, and fixed endpoint vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^m$.

1. Compute $\mathbf{X}$ from Equation (18). Find prime-leading representatives $(F_i)_i$ and compute T and B exactly as in the preprocessing phase of [Protocol 6.1](#).
2. Derive $A_T, A_B, \mathbf{H}_-, \mathbf{H}_+$ independently from public coins, where each $\mathbf{H}_{\pm}$ contains seven bases. Compute

$$C_{\text{fix}} := T \cdot A_T + B \cdot A_B, \quad A_- := -\langle \mathbf{a}, \mathbf{G} \rangle, \quad A_+ := \langle \mathbf{b}, \mathbf{G} \rangle.$$

Validate the complete preprocessing record and output

$$\text{pk}_{\text{Range}} := (\mathbf{a}, \mathbf{b}, \mathbf{X}, (F_i)_i, T, B),$$

$$\text{vk}_{\text{Range}} := (\text{gk}, \mathbf{G}, A_T, A_B, \mathbf{H}_-, \mathbf{H}_+, C_{\text{fix}}, A_-, A_+).$$

Common input: vk_{Range} and $V \in \mathcal{C}_{\Delta}$.

Prover input: pk_{Range} and $\mathbf{v} \in \mathbb{Z}^m$ with $a_j \leq v_j \leq b_j$ for every $j \in [m]$ and $V = \langle \mathbf{v}, \mathbf{G} \rangle$.

1. Set $\mathbf{z}_- := \mathbf{v} - \mathbf{a}$ and $\mathbf{z}_+ := \mathbf{b} - \mathbf{v}$, and let $I_-, I_+ \subseteq [n]$ collect the binary supports of their coordinates within the corresponding blocks of $\mathbf{X}$. Apply the online reduction of [Protocol B.2](#) with $d = 2$ to obtain witnesses $\mathbf{w}_-, \mathbf{w}_+$ for

$$Y_- = V + A_-, \quad Y_+ = A_+ - V,$$

and send the single combined commitment

$$U := \langle \mathbf{w}_-, \mathbf{H}_- \rangle + \langle \mathbf{w}_+, \mathbf{H}_+ \rangle.$$

2. Run [Protocol 5.1](#) with $k = 2$ on commitments (C_{fix}, U) , witness $(T, B, \mathbf{w}_-, \mathbf{w}_+)$, and predicate $\Psi_{V, \mathbf{a}, \mathbf{b}} = 0$. The verifier accepts exactly when this invocation accepts.

Corollary 6.5 (Batched class group range argument). *For independently sampled bases $\mathbf{G}$, under DMC and the discrete-log assumption, [Protocol 6.2](#) is a public-coin argument of knowledge for $\mathcal{R}_{\text{Range}}^{\mathbf{a}, \mathbf{b}}$ of Section 3.4. It is not zero knowledge. Write $n = \sum_j n_j$, let $\rho(\lambda) = |\Delta|_{\text{bit}} + O(\log|\Delta|_{\text{bit}})$ be the prime-representative bound of Section 6.1.2, and let*

$$s(n, \lambda) = O(n\rho(\lambda) + |\Delta|_{\text{bit}} + \lambda)$$

bound the honest witness coordinates. Preprocessing costs n prime-representative searches and $\tilde{O}(n\rho(\lambda))$ bit operations. The prover performs a constant number of MSMs, with scalars of $O(s(n, \lambda) + \lambda)$ bits, and $\tilde{O}(s(n, \lambda))$ bit operations. The verifier performs one 19-term MSM, with scalars of at most $4\lambda + 1$ bits, and $O(M(|\Delta|_{\text{bit}}) + M(\lambda))$ bit operations. The proof is two group elements and 16 elements of $\mathbb{F}_\ell$, independently of m . For fixed λ the prover therefore runs a constant number of MSMs with scalars of $O(n)$ bits and $O(M(n) \log n)$ bit operations. The exact constants and the proof appear in Appendix C.

When every interval is a full binary range anchored at zero, the upper target carries no information and can be dropped.

Corollary 6.6 (Power-of-two ranges). *Suppose $\mathbf{a} = \mathbf{0}$ and $b_j + 1 = 2^{n_j}$ for every j . Then Protocol 6.2 run with the single target $Y_- = V$ and predicate Φ_V is already a public-coin argument of knowledge for $\mathcal{R}_{\text{Range}}^{\mathbf{0}, \mathbf{b}}$. Its witness is one seven-coordinate block instead of two, and the preprocessing drops A_- , A_+ and $\mathbf{H}_+$.*

Proof. A subset of $\mathbf{X}$ encodes exactly the vectors $\mathbf{z}$ with $0 \leq z_j < 2^{n_j} = b_j + 1$, so the upper bound is the subset-product statement itself and Y_+ constrains nothing. Formally, extraction from the single target yields $I \subseteq [n]$, and partitioning it by base gives $\mathbf{z}$ with $0 \leq z_j \leq b_j$ and $V = \langle \mathbf{z}, \mathbf{G} \rangle$. Two distinct openings of V by polynomially bounded integer vectors would break the multi-generator discrete-log assumption, so $\mathbf{z}$ is the opening of V and is a valid witness. Completeness and the costs are those of Corollary 6.5 with one target in place of two. $\square$

7 The Mist Polynomial Commitment

This section instantiates the DewTwo PCS framework with the class-group arguments developed above. We first define the underlying commitment scheme and then give the weak evaluation protocol. The main changes to DewTwo’s protocol is that its bounded-integer checks are discharged by the batched range argument of Section 6.3.

7.1 Integer encoding and the commitment scheme

Fix a prime q , an integer $\mu \geq 4$, and $N := 2^\mu$. For $p \in \mathbb{F}_q^{\leq 1}[X_1, \dots, X_\mu]$, let $\tilde{\mathbf{p}} = (\tilde{p}_0, \dots, \tilde{p}_{N-1}) \in [0, q-1]^N$ be the canonical integer lift of its coefficient vector. For an integer base $\alpha > q$, define the *polynomial integer encoding*

$$\text{Enc}_\alpha(p) := \sum_{i=0}^{N-1} \tilde{p}_i \alpha^i \in \mathbb{Z}. \quad (20)$$

The base- α representation shows immediately that this map is injective.

We use the same bounded opening hint space as DewTwo [BMSS25]. Put

$$\begin{aligned} A &:= Nq^{2\mu+1} + 1, & B &:= N^{8\mu}(2\mu)^\lambda, \\ \mathcal{H} &:= \left\{ \frac{a}{b} \in \mathbb{Q} : \left| \frac{a}{b} \right| \leq A, \ 0 < b \leq B \right\}. \end{aligned}$$

Choose a power $\alpha = q^k$ satisfying $\alpha > 2AB^2$. In the parameter regime $\lambda < \log q$ and $4 \leq \mu \leq \lambda/4$, one may use DewTwo’s choice $\alpha = q^{7\mu}$.

The commitment scheme $\text{CS}_{\text{Mist}} = (\text{Setup}_{\text{Mist}}, \text{Commit}_{\text{Mist}}, \text{Open}_{\text{Mist}})$ is as follows. Setup samples $\text{gk} \leftarrow \text{CGGen}(1^\lambda)$ and an independent $G \leftarrow \mathcal{C}_\Delta$, and publishes $(\text{gk}, G, q, \mu, \alpha, \mathcal{H})$. To commit to p ,

compute

$$C := \text{Enc}_\alpha(p) \cdot G \quad (21)$$

and output (C, d) with $d := \tilde{\mathbf{p}}$. More generally, $\text{Open}_{\text{Mist}}(\mathbf{pp}, C, p, d) = 1$ for $d = (d_0, \dots, d_{N-1}) \in \mathcal{H}^N$ exactly when the committed value $c := \sum_{i=0}^{N-1} d_i \alpha^i$ lies in $\mathbb{Z}[1/2]$ and

$$C = c \cdot G, \quad d_i \bmod q = p_i \quad (i = 0, \dots, N-1). \quad (22)$$

The opening space is DewTwo's unchanged: the recovery argument of Section 7.2 returns coefficients that are bounded rationals, not dyadic ones, and their boundedness is what gives binding. What the extractors of Section 5 contribute is dyadic values for the *committed integers*, which is why the sum is required to lie in $\mathbb{Z}[1/2]$, where the scalar action is defined; honest commitments use the canonical integer lift.

Lemma 7.1 (Binding of CS_{Mist}). *Under DLOG, CS_{Mist} is binding.*

The proof appears in Appendix C.

7.2 Weak evaluation

We follow DewTwo's coefficient splitting. If $c = ab = \sum_{j=0}^m c_j \alpha^j$ is the canonical base- α representation of the nonnegative integer ab , the algorithm separates the digits below, at, and above position i .

Algorithm 7.1: CoeffSplit(i, a, b, α)

Input: An index $i \in \mathbb{N}$, nonnegative integers a, b , and an integer base $\alpha \geq 2$.

Output: Integers (s, t, u) such that $ab = s + t\alpha^i + u\alpha^{i+1}$.

Compute $c := ab$ and its base- α digits $c = \sum_{j=0}^m c_j \alpha^j$, with $0 \leq c_j < \alpha$. Return

$$s := \sum_{j=0}^{i-1} c_j \alpha^j, \quad t := c_i, \quad u := \sum_{j=i+1}^m c_j \alpha^{j-i-1}.$$

Let $\tilde{y} \in [0, q-1]$ denote the canonical lift of $y \in \mathbb{F}_q$. For $\mathbf{z} \in \mathbb{F}_q^\mu$, lift each coordinate to $\tilde{z}_j \in [0, q-1]$ and form the multilinear monomials *over the integers*,

$$\tilde{\chi}_i(\mathbf{z}) := \prod_{j=1}^{\mu} \tilde{z}_j^{\text{bin}(i,j)} \in \mathbb{Z}, \quad 0 \leq \tilde{\chi}_i(\mathbf{z}) < q^\mu,$$

which reduce to $\chi_i(\mathbf{z})$ modulo q but are not themselves reduced. Define the reverse encoding

$$\sigma_{\mathbf{z}} := \sum_{i=0}^{N-1} \tilde{\chi}_{N-1-i}(\mathbf{z}) \alpha^i = \prod_{j=1}^{\mu} \left(\tilde{z}_j + \alpha^{2^{\mu-j}} \right), \quad (23)$$

which is what forces the monomials to be lifted unreduced: the identity holds over $\mathbb{Z}$, and the digit $\tilde{\chi}_{N-1-i}(\mathbf{z})$ in position i is bounded by $q^\mu < \alpha$ so the expansion really is the base- α one. The verifier therefore never constructs this $\Theta(N \log \alpha)$ -bit integer, and obtains $\sigma_{\mathbf{z}} \bmod \ell$ in $O(\mu)$ field operations by reducing Equation (23).

The coefficient estimates naturally give the strict DewTwo thresholds

$$S_\star := Nq^{\mu+1}\alpha^{N-2}, \quad T_\star := Nq^{\mu+1}, \quad U_\star := q^{\mu+1}\alpha^{N-2}, \quad (24)$$

with $0 \leq s < S_\star$, $0 \leq t < T_\star$, and $0 \leq u < U_\star$. To use exact binary power families for our range proofs, round each threshold up and put

$$\begin{aligned} B_S &:= 2^{\lceil \log_2 S_\star \rceil}, & B_T &:= 2^{\lceil \log_2 T_\star \rceil}, \\ B_U &:= 2^{\lceil \log_2 U_\star \rceil}, & \mathbf{B} &:= (B_S, B_T, B_U). \end{aligned} \quad (25)$$

The range layer proves $(s, t, u) \in [0, \mathbf{B} - 1]$ using power families of lengths exactly $\log_2 B_S$, $\log_2 B_T$, and $\log_2 B_U$. Now, for variables (c, w, s, t, u) define the arithmetic predicate

$$g_{\mathbf{z}, y}(c, w, s, t, u) := (t - \tilde{y} - qw)^2 + (\sigma_{\mathbf{z}}c - s - t\alpha^{N-1} - u\alpha^N)^2. \quad (26)$$

Protocol 7.1: Reduction Π_{WeakEval}

Common input: Public parameters for CS_{Mist} , independent bases $H_w, G_s, G_t, G_u \in \mathcal{C}_\Delta$, and an evaluation instance $(q, \mu, C, \mathbf{z}, y)$ in which $\mathbf{z}$ is sampled independently of C .

Prover input: A valid opening (p, d) of C with canonical integer $d = \tilde{\mathbf{p}}$ and $p(\mathbf{z}) = y$.

1. Set $c := \text{Enc}_\alpha(p)$ and compute

$$(s, t, u) := \text{CoeffSplit}(N - 1, c, \sigma_{\mathbf{z}}, \alpha), \quad w := (t - \tilde{y})/q.$$

2. Send

$$W := w \cdot H_w, \quad V := s \cdot G_s + t \cdot G_t + u \cdot G_u.$$

3. Output the product of the following target instances, together with the shared witness (c, w, s, t, u) :

- (a) the three-block PoKEMath instance with commitments (C, W, V) , respective bases (G) , (H_w) , and (G_s, G_t, G_u) , and predicate $g_{\mathbf{z}, y} = 0$;
- (b) the batched range instance

$$(V, (s, t, u)) \in \mathcal{R}_{\text{Range}}^{\mathbf{0}, \mathbf{B}-1}.$$

Theorem 7.2 (Weak evaluation). *Under the discrete-log assumption, and for an evaluation point $\mathbf{z} \leftarrow \mathbb{F}_q^\mu$ sampled uniformly, Protocol 7.1 is a weak reduction of knowledge*

$$\mathcal{R}_{\text{Eval}} \longrightarrow \mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 3} \times \mathcal{R}_{\text{Range}}^{\mathbf{0}, \mathbf{B}-1}.$$

Its proof appears in Appendix C; the coefficient-recovery step is the algebraic argument of DewTwo [BMSS25], unchanged after allowing dyadic extracted openings.

For the final protocol we fuse the range reduction with the arithmetic check. The bounds of Equation (25) are powers of two and the ranges start at zero, so Corollary 6.6 applies: the range layer is the single target $Y = V$ with witness $\mathbf{w}$, and no upper target is needed. The commitment W to w and this range commitment are both purely auxiliary, so we concatenate their bases and replace them by one commitment U_A with 8 coordinates. The commitments C , V , and C_{fix} remain separate: they respectively bind the encoded polynomial, the range target (s, t, u) , and the fixed preprocessed pair (T, B) . The four opening blocks have dimensions 1, 3, 2, $(1 + 7 = 8)$, for 14 witness coordinates in total.

We invoke the multi-commitment protocol of Section 5.2 with $k = 4$, unrolled below so that the complete argument can be read in one place. The Vandermonde extraction in Theorem 5.1 recovers the individual opening equations; the ordering requirement is highlighted in Remark 5.2.

Protocol 7.2: Optimized argument WeakEval^{opt}

Common and prover inputs: As in Protocol 7.1, together with the preprocessing for the batched range $[0, B - 1]$.

1. Compute (c, w, s, t, u) and V exactly as in Protocol 7.1. Apply the reduction underlying Protocol 6.2 to (s, t, u) , in the single-target form of Corollary 6.6, to obtain $\mathbf{w}$. Send V and the single auxiliary commitment

$$U_A := w \cdot H_w + \langle \mathbf{w}, \mathbf{H} \rangle.$$

2. Both parties form the four blocks

$$\begin{aligned} (\mathbf{G}_1, U_1, \mathbf{u}_1) &:= ((G), C, (c)), \\ (\mathbf{G}_2, U_2, \mathbf{u}_2) &:= ((G_s, G_t, G_u), V, (s, t, u)), \\ (\mathbf{G}_3, U_3, \mathbf{u}_3) &:= ((A_T, A_B), C_{\text{fix}}, (T, B)), \\ (\mathbf{G}_4, U_4, \mathbf{u}_4) &:= ((H_w, \mathbf{H}), U_A, (w, \mathbf{w})), \end{aligned}$$

and the joint predicate

$$f := g_{\mathbf{z}, y}(c, w, s, t, u) + \Phi_V(\tau, \beta, \mathbf{w}).$$

3. The verifier samples $\ell \leftarrow \text{Primes}(2\lambda)$ and sends it to the prover.
4. For every $j \in [4]$ the prover computes

$$\mathbf{r}_j := \mathbf{u}_j \bmod \ell, \quad \mathbf{q}_j := \frac{\mathbf{u}_j - \mathbf{r}_j}{\ell},$$

and sends the 14 residues $(\mathbf{r}_j)_{j \in [4]}$.

5. After the residues are fixed, the verifier samples $\eta \leftarrow \mathbb{F}_\ell$ and sends it; both parties set ω_j to the canonical integer representative of η^{j-1} .
6. The prover sends the single quotient element

$$Q := \sum_{j=1}^4 \omega_j \langle \mathbf{q}_j, \mathbf{G}_j \rangle.$$

The verifier accepts if

$$\begin{aligned} \sum_{j=1}^4 \omega_j U_j &= \ell \cdot Q + \sum_{j=1}^4 \omega_j \langle \mathbf{r}_j, \mathbf{G}_j \rangle, \\ f(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4) &= 0 \pmod{\ell}. \end{aligned}$$

Corollary 7.3 (Optimized weak evaluation). *Under DMC and DLOG, Protocol 7.2 is a public-coin weak evaluation argument of knowledge for $\mathcal{R}_{\text{Eval}}$. Let*

$$L := N \log \alpha, \quad n_R := \log_2 B_S + \log_2 B_T + \log_2 B_U,$$

and let $s_R = s(n_R, \lambda)$ be the range-witness bit bound from Corollary 6.5. Then:

- Preprocessing: the range layer contains n_R power-family elements, so by Corollary 6.5 it costs n_R prime-representative searches (Section 6.1.2) together with $\tilde{O}(n_R \rho(\lambda))$ bit operations for the product and CRT trees.
- Prover: monomial expansion and integer encoding take $\tilde{O}(L)$ bit operations, CoeffSplit takes $O(M(L))$ bit operations, and the range-witness construction takes $O(M(s_R)(\log(n_R + 1) + \log s_R))$

bit operations. Group work is

$$\text{MSM}(3, L) + \text{MSM}(8, \max\{L, s_R\}) + \text{MSM}(14, \max\{L, s_R\} + 2\lambda).$$

Hence the prover is quasilinear in N for fixed security parameters.

- Verifier: Equation (23) and the joint predicate require $O(\mu)$ field operations. The batched opening check uses one 19-term MSM with at most $(4\lambda + 1)$ -bit scalars, plus constantly many group additions. Thus verifier work is $O(\log N)$.
- Proof size: the prover sends V and U_A , then the 14 residues and one quotient commitment. The proof therefore contains exactly three class group elements and 14 field elements. At $\lambda = 128$ with the 288-byte compressed class group representation of Section 4.1, this is

$$3 \cdot 288 + 14 \cdot 32 = 1312 \text{ bytes.}$$

Proof. Apply Theorem 7.2 and substitute the range reduction of Corollary 6.6, which applies because the bounds of Equation (25) are powers of two and the ranges start at zero. Then apply Theorem 5.1 to the four opening blocks. The coordinate vectors formerly committed by W and the range commitment are a direct sum under disjoint independent bases, so replacing those two elements by U_A preserves their joint opening relation. The coordinate count is $1 + 3 + 2 + 8 = 14$, while the protocol retains one quotient for the batched opening. Together with V, U_A , this gives the stated communication. The remaining bounds follow from Corollary 6.5, the cost of `CoeffSplit`, and Theorem 5.1. $\square$

7.3 Arbitrary evaluation points

Protocol 7.2 is sound for an evaluation point sampled independently of the commitment, which is the only place the extraction argument needs randomness. Following [BMSS25], a full evaluation argument runs the weak protocol twice: once at a verifier-sampled random point $\mathbf{r}$, which forces the commitment to be well formed, and once at the claimed point $\mathbf{z}$, which establishes the claimed value. Knowledge soundness for adaptively chosen points follows from their analysis.

The commitment C and the preprocessed pair C_{fix} are common to both and the two range checks carry the same bounds $\mathbf{B}$, so they merge into one batched range over the six coordinates $(s_{\mathbf{r}}, t_{\mathbf{r}}, u_{\mathbf{r}}, s_{\mathbf{z}}, t_{\mathbf{z}}, u_{\mathbf{z}})$ – precisely what Corollary 6.5 supplies, and the reason the range layer was stated for arbitrary m . One BatchPoKEMath quotient then covers every block. The opening blocks become 1, 6, 2, $(2 + 7 = 9)$, for 18 witness coordinates, and the transcript is still the three class-group elements V, U_A and the quotient, together with the value $y_{\mathbf{r}} \in \mathbb{F}_q$ that the prover claims at the random point. At $\lambda = 128$, and with q of 256 bits, this is $3 \cdot 288 + 18 \cdot 32 + 32 = 1472$ bytes.

References

- [ACFY25] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: Reed-solomon proximity testing with super-fast verification. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 214–243. Springer, Cham, May 2025.
- [AGK⁺26] Alexander Abdugafarov, Albert Garreta, Amit Kumar, Michał Osadnik, Psi Vesely, Ilia Vlasov, and Kai Zhe Zheng. Zinc+: SNARKs for polynomial rings. Cryptology ePrint Archive, Paper 2026/855, 2026.

- [AGL⁺23] Arasu Arun, Chaya Ganesh, Satya V. Lokam, Tushar Mopuri, and Sriram Sridhar. Dew: A transparent constant-sized polynomial commitment scheme. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 542–571. Springer, Cham, May 2023.
- [BBB⁺18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.
- [BBF19] Dan Boneh, Benedikt Bünz, and Ben Fisch. Batching techniques for accumulators with applications to IOPs and stateless blockchains. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 561–586. Springer, Cham, August 2019.
- [BCG⁺14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014.
- [BCIL23] Cyril Bouvier, Guilhem Castagnos, Laurent Imbert, and Fabien Laguillaumie. I want to ride my BICYCL : BICYCL implements CryptographY in C++ groups. *Journal of Cryptology*, 36(3):17, July 2023.
- [BF97] Dan Boneh and Matthew K. Franklin. Efficient generation of shared RSA keys (extended abstract). In Burton S. Kaliski, Jr., editor, *CRYPTO’97*, volume 1294 of *LNCS*, pages 425–439. Springer, Berlin, Heidelberg, August 1997.
- [BF23] Benedikt Bünz and Ben Fisch. Multilinear schwartz-zippel mod N and lattice-based succinct arguments. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part III*, volume 14371 of *LNCS*, pages 394–423. Springer, Cham, November / December 2023.
- [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.
- [BMSS25] Benedikt Bünz, Tushar Mopuri, Alireza Shirzad, and Sriram Sridhar. DewTwo: A transparent PCS with quasi-linear prover, logarithmic verifier and 4.5KB proofs from falsifiable assumptions. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 549–583. Springer, Cham, August 2025.
- [BP97] Niko Bari and Birgit Pfizmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *EUROCRYPT’97*, volume 1233 of *LNCS*, pages 480–494. Springer, Berlin, Heidelberg, May 1997.
- [BS96] Wieb Bosma and Peter Stevenhagen. On the computation of quadratic 2-class groups. *Journal de Théorie des Nombres de Bordeaux*, 8(2):283–313, 1996.
- [BV07] Johannes Buchmann and Ulrich Vollmer. *Binary Quadratic Forms: An Algorithmic Approach*, volume 20 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, Heidelberg, 2007.
- [BW88] Johannes Buchmann and Hugh C. Williams. A key-exchange system based on imaginary quadratic fields. *Journal of Cryptology*, 1(2):107–118, June 1988.

- [CHA24] Matteo Campanelli and Mathias Hall-Andersen. Fully succinct arguments over the integers from first principles. Cryptology ePrint Archive, Report 2024/1548, 2024.
- [CHI⁺21] Megan Chen, Carmit Hazay, Yuval Ishai, Yuriy Kashnikov, Daniele Micciancio, Tarik Riviere, abhi shelat, Muthu Venkitasubramaniam, and Ruihan Wang. Diogenes: Lightweight scalable RSA modulus generation with a dishonest majority. In *2021 IEEE Symposium on Security and Privacy*, pages 590–607. IEEE Computer Society Press, May 2021.
- [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.
- [CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, Berlin, Heidelberg, August 2002.
- [Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, Heidelberg, 1993.
- [CXNB26] Jessica Chen, Lucas Xia, Wilson Nguyen, and Benedikt Bünz. Limber: Low overhead SNARKs for integers from any PCS. Cryptology ePrint Archive, Paper 2026/1635, 2026.
- [DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 125–142. Springer, Berlin, Heidelberg, December 2002.
- [DGS20] Samuel Dobson, Steven D. Galbraith, and Benjamin Smith. Trustless groups of unknown order with hyperelliptic curves. Cryptology ePrint Archive, Report 2020/196, 2020.
- [DK02] Ivan Damgård and Maciej Koprowski. Generic lower bounds for root extraction and signature schemes in general groups. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 256–271. Springer, Berlin, Heidelberg, April / May 2002.
- [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.
- [Fu25] Shihui Fu. Improved constant-sized polynomial commitment schemes without trusted setup. *CiC*, 2(2):27, 2025.
- [GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Berlin, Heidelberg, May 2013.
- [GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 113–122. ACM Press, May 2008.
- [GLS⁺23] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic SNARKs for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 193–226. Springer, Cham, August 2023.

- [GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.
- [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.
- [GWV⁺25] Albert Garreta, Hendrik Waldner, Ilia Vlasov, Katerina Hristova, Luca Dall’Ava, Marko Cupic, and Matthew Klein. Zinc: Succinct arguments with small arithmetization overheads from IOPs of proximity to the integers. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 259–291. Springer, Cham, August 2025.
- [Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.
- [KP23] Abhiram Kothapalli and Bryan Parno. Algebraic reductions of knowledge. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 669–701. Springer, Cham, August 2023.
- [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.
- [Lee20] Jonathan Lee. The security of groups of unknown order based on Jacobians of hyperelliptic curves. Cryptology ePrint Archive, Report 2020/289, 2020.
- [LXZ21] Tianyi Liu, Xiang Xie, and Yupeng Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2968–2985. ACM Press, November 2021.
- [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.
- [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.
- [PTX26] Dimitrios Papadopoulos, Qiang Tang, and Jiajun Xin. Morphic accumulators and applications. In *Advances in Cryptology – CRYPTO 2026: 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–20, 2026, Proceedings, Part IX*, page 460–491, Berlin, Heidelberg, 2026. Springer-Verlag.
- [RIS26] RISC Zero. RISC Zero: A zero-knowledge verifiable general computing platform. <https://github.com/risc0/risc0>, 2026.
- [RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, February 1978.
- [SB25] István András Seres and Péter Bursi. Behemoth: Transparent polynomial commitment scheme with constant opening proof size and verifier time. In Abderrahmane Nitaj,

- Svetla Petkova-Nikova, and Vincent Rijmen, editors, *AFRICACRYPT 25*, volume 15651 of *LNCS*, pages 343–369. Springer, Cham, July 2025.
- [SBK24] István András Seres, Péter Burcsi, and Péter Kutas. How (not) to hash into class groups of imaginary quadratic fields? Cryptology ePrint Archive, Report 2024/034, 2024.
- [SSPP26] Alireza Shirzad, Sriram Sridhar, Dimitrios Papadopoulos, and Charalampos Papamanthou. Relaxed modular PCS from arbitrary PCS and applications to SNARKs for integers. Cryptology ePrint Archive, Paper 2026/347, 2026.
- [Suc26] Succinct Labs. SP1: A performant, 100% open-source, contributor-friendly zkVM. <https://github.com/succinctlabs/sp1>, 2026.
- [TCLM21] Sri Aravinda Krishnan Thyagarajan, Guilhem Castagnos, Fabien Laguillaumie, and Giulio Malavolta. Efficient CCA timed commitments in class groups. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2663–2684. ACM Press, November 2021.
- [Wes19] Benjamin Wesolowski. Efficient verifiable delay functions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 379–407. Springer, Cham, May 2019.

A Additional Preliminaries

A.1 Root-aware generic group model for groups of unknown order

We use the generic group model for groups of unknown order [DK02, BBF19], augmented with the operation that is efficiently available in our class group interface. As in DewTwo, the unknown order N is sampled uniformly from a public interval $[N_\lambda^-, N_\lambda^+]$ of odd integers, where $1/N_\lambda^-$ and $1/(N_\lambda^+ - N_\lambda^-)$ are negligible. The generic group is the additive cyclic group $\mathbb{Z}_N$, exposed through a random injective encoding

$$\sigma : \mathbb{Z}_N \longrightarrow \{0, 1\}^\rho.$$

The adversary receives only encodings and the public lower and upper bounds on N , not the sampled order itself. We write $\mathbb{G}$ for the encoded group. The adversary accesses the following oracles:

$$\begin{aligned} \mathcal{O}_{\text{samp}}() : & \quad X \leftarrow \mathbb{Z}_N; \text{ return } \sigma(X), \\ \mathcal{O}_{\text{grp}}(\sigma(X), \sigma(Y), \pm) : & \quad \text{return } \sigma(X \pm Y \bmod N), \\ \mathcal{O}_{1/2}(\sigma(X)) : & \quad \text{return } \sigma(\text{Half}(X)). \end{aligned}$$

Here $\text{Half}(X)$ is the unique $Y \in \mathbb{G}$ satisfying $2 \cdot Y = X$; it exists because $|\mathbb{G}|$ is odd. Equality is tested only by comparing labels. Labels are sampled lazily and consistently with the oracle answers. We call this the *root-aware GGM* and denote it by $\text{GGM}_{1/2}$. It is an idealized model, not a falsifiable assumption.

The usual symbolic representation lemma now tracks coefficients in $\mathbb{Z}[1/2]$: addition and subtraction add coefficient vectors, while a halving query divides one by two. Consequently, an element fixed after T oracle calls has denominator depth at most T and numerators bounded by 2^{T+2} . This is the source of the parameters d_T and B_T in Definition 4.2.

A.2 Class group algorithms

Reduction of a binary quadratic form, used as $\text{red}(\cdot)$ throughout Sections 4.2 and 6.

Algorithm A.1: Reduce(f)

Input: A form $f = (a, b, c)$ of discriminant Δ .

Output: The reduced form $\bar{f}$ and $M \in \text{SL}_2(\mathbb{Z})$ such that $\bar{f} = f^M$.

Set $M \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Repeat:

1. Set $q \leftarrow \lfloor (a - b)/(2a) \rfloor$ and

$$(a, b, c) \leftarrow (a, b + 2qa, aq^2 + bq + c), \quad M \leftarrow M \begin{pmatrix} 1 & q \\ 0 & 1 \end{pmatrix}.$$

2. If $a > c$, or if $a = c$ and $b < 0$, set

$$(a, b, c) \leftarrow (c, -b, a), \quad M \leftarrow M \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

and repeat. Otherwise return $((a, b, c), M)$.

A.3 Reductions of knowledge

We spell out the three properties of a reduction of knowledge $\Pi = (\text{Setup}, \mathcal{P}, \mathcal{V})$ from $\mathcal{R}_1$ to $\mathcal{R}_2$ summarized in Section 3.2.

Completeness For every $N \in \mathbb{N}$, every public parameter $\text{pp} \leftarrow \text{Setup}(1^\lambda, N)$, and every $(\text{pp}, x_1, w_1) \in \mathcal{R}_1$, an honest interaction outputs (x_2, w_2) such that $(\text{pp}, x_2, w_2) \in \mathcal{R}_2$ with probability one.

Knowledge soundness For every $N \in \mathbb{N}$ and every expected polynomial-time stateful adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists an expected polynomial-time extractor $\mathcal{E}$ given oracle access to $\mathcal{A}$ such that

$$\Pr \left[\begin{array}{l} (\text{pp}, x_2, w_2) \in \mathcal{R}_2 \\ \wedge (\text{pp}, x_1, w_1) \notin \mathcal{R}_1 \end{array} \middle| \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda, N); \\ (x_1, \text{st}) \leftarrow \mathcal{A}_1(\text{pp}); \\ (x_2, w_2) \leftarrow \langle \mathcal{A}_2(\text{st}), \mathcal{V}(\text{pp}, x_1) \rangle; \\ w_1 \leftarrow \mathcal{E}^{\mathcal{A}}(\text{pp}, x_1) \end{array} \right] \leq \text{negl}(\lambda).$$

Public reducibility There is a deterministic polynomial-time function φ such that for every interaction between a stateful prover and $\mathcal{V}$ with transcript τ and target instance x_2 ,

$$\varphi(\text{pp}, x_1, \tau) = x_2.$$

We write $\Pi : \mathcal{R}_1 \rightarrow \mathcal{R}_2$ when Π satisfies these properties.

Theorem A.1 (Sequential composition [KP23]). *Let $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ be ternary relations. If $\Pi_1 : \mathcal{R}_1 \rightarrow \mathcal{R}_2$ and $\Pi_2 : \mathcal{R}_2 \rightarrow \mathcal{R}_3$ are reductions of knowledge with the same setup algorithm, then their sequential composition $\Pi_2 \circ \Pi_1$ is a reduction of knowledge from $\mathcal{R}_1$ to $\mathcal{R}_3$.*

A.4 Modular consistency

The assumption of DewTwo [BMSS25] summarized in Section 4.3, written out.

Definition A.2 (Modular consistency [BMSS25]). Let $\text{rep} = \text{poly}(\lambda)$ be the representation size of a group element. MCA holds if, for every $n, T = \text{poly}(\lambda)$, every $(T \cdot \text{rep})$ -step stateful Turing machine

$\mathcal{A}_1$, and every PPT $\mathcal{A}_2$,

$$\text{Adv}_{\mathcal{A},T,n}^{\text{MC}} := \Pr \left[\begin{array}{c} \forall i \in [T] : U = \langle \mathbf{r}_i, \mathbf{G} \rangle + \ell_i \cdot Q_i \\ \wedge \langle \mathbf{x}, \mathbf{G} \rangle \neq U \end{array} \mid \begin{array}{l} \mathbf{gk} \leftarrow \text{CGGen}(1^\lambda); \\ \mathbf{G} \leftarrow \mathbb{G}^n; \\ (U, \text{st}) \leftarrow \mathcal{A}_1(\mathbf{gk}, \mathbf{G}); \\ [(\mathbf{r}_i, Q_i, \ell_i)]_{i=1}^T \leftarrow \mathcal{A}_2(\text{st}); \\ \ell_1, \dots, \ell_T \text{ pairwise coprime}; \\ x_j \leftarrow \text{CRT}((r_{i,j})_{i=1}^T, (\ell_i)_{i=1}^T); \\ j \in [n] \end{array} \right] \leq \text{negl}(\lambda).$$

A.5 Reductions between assumptions

The following statements correspond to the arrows in Figure 1.

Lemma A.3 (MCA implies DMC). *For every polynomial parameter tuple (T, n, s) ,*

$$\text{MCA}_{T,n,s} \implies \text{DMC}_{T,n,s}.$$

In particular, the reduction has no parameter or advantage loss: the ordinary integer CRT lift used by MCA belongs to the DMC candidate set.

Proof. Let $\mathbf{z}$ be the coordinate-wise integer CRT lift of the responses. By Equation (3), $\mathbf{z} \in \mathcal{X}_{d_T, \ell}((\mathbf{r}_i)_i)$. Thus every winning DMC transcript also satisfies $U \neq \langle \mathbf{z}, \mathbf{G} \rangle$ and is a winning MCA transcript. The identity reduction gives $\text{Adv}^{\text{DMC}} \leq \text{Adv}^{\text{MC}}$ with the same parameters. $\square$

Lemma A.4 (DMC implies adaptive root). *If DMC holds for all polynomial (T, n, s) , then the adaptive-root property AR holds for challenges in $\text{Primes}(2\lambda)$.*

Proof. Suppose an adaptive-root adversary fixes $U \neq 0$ and, after seeing a random prime ℓ , returns X such that $\ell X = U$ with non-negligible probability. Rewind its second stage and collect successful answers X_i for s distinct primes ℓ_i , where s is polynomial and $\prod_i \ell_i > 2B_T 2^{d_T}$. The standard heavy-row argument makes this expected polynomial time. Give DMC the transcripts $(\mathbf{0}, X_i, \ell_i)$. They are accepting, while their candidate set is the singleton $\{\mathbf{0}\}$; hence no candidate opens the nonzero element U . This contradicts DMC. $\square$

Lemma A.5 (Adaptive root implies low order). *If AR holds, then the low-order property LO holds.*

Proof. Suppose (W, m) satisfies $W \neq 0$, $m \neq 0$, and $mW = 0$. Fix $U := W$ before receiving the adaptive-root challenge ℓ . Except with negligible probability, $\gcd(\ell, m) = 1$: a polynomial-size nonzero m has only polynomially many prime divisors. Choose $a, b \in \mathbb{Z}$ with $a\ell + bm = 1$ and output $X := aW$. Then $\ell X = (1 - bm)W = W = U$, contradicting adaptive-root security. $\square$

Lemma A.6 (DMC in the root-aware GGM). *For the bounded-order sampler above, DMC holds in $\text{GGM}_{1/2}$ for every polynomial (T, n, s) satisfying the challenge-product condition in Definition 4.2.*

Proof. Use the usual lazy-sampling representation argument, now with coefficients in $\mathbb{Z}[1/2]$. Except with negligible generic-collision probability, every label has a unique formal linear representation in the initially sampled generators. An element U fixed after T calls has denominator depth at most d_T , so, clearing denominators, it has a representation $U = \sum_j (a_j/2^{d_T}) \cdot G_j$ over the public $\mathbf{G}$ and any auxiliary sampled generators, with $|a_j| \leq B_T 2^{d_T}$.

Each accepting transcript implies, coefficient-wise, that the numerators a_j of the $\mathbf{G}$ -coordinates are congruent to $2^{d_T} r_{i,j}$ modulo ℓ_i , while every auxiliary numerator is congruent to 0 modulo ℓ_i . Since $L = \prod_i \ell_i > 2B_T 2^{d_T}$ and every such numerator has absolute value at most $B_T 2^{d_T} < L/2$, two of them congruent modulo L must be equal (their difference is a multiple of L of absolute value less than L , hence zero). Consequently every auxiliary numerator vanishes exactly, and the numerator vector of the $\mathbf{G}$ -coordinates is exactly $\text{CRT}^\pm((2^{d_T} r_{i,j})_i, (\ell_i)_i)$, the $e = d_T$ term of Equation (3). So the coefficient vector $2^{-d_T} \text{CRT}^\pm((2^{d_T} r_{i,j})_i, (\ell_i)_i)$ lies in $\mathcal{X}_{d_T, \ell}((\mathbf{r}_i)_i)$ and opens U , so the DMC winning condition is impossible outside the negligible generic-collision event under the stated order bounds. $\square$

Lemma A.7 (Discrete logarithm in the root-aware GGM). *For independently sampled bases and polynomially bounded representations, the discrete-log assumption holds in $\text{GGM}_{1/2}$.*

Proof. Independently sampled bases are distinct formal generators. Two different polynomially bounded integer vectors therefore give different formal linear representations; equality of their labels is a generic collision, whose probability is negligible under the stated order bounds. The same conclusion holds for polynomial-depth dyadic vectors after clearing a common power-of-two denominator. $\square$

B Deferred Protocols and Algorithms

B.1 Dyadic PoKE

The proof of knowledge of exponent protocol (PoKE) [BBF19] lets a prover demonstrate knowledge of the exponents in a multi-base commitment. The public parameters are $\mathbf{pp} := (\mathbf{gk}, \mathbf{G})$ as produced by Algorithm 4.2, so that in particular the bases are sampled by the setup rather than chosen by the prover. Consider the relation

$$\mathcal{R}_{\text{PoKE}}^{\mathbb{Z}} := \{((\mathbf{gk}, \mathbf{G}), U, \mathbf{u}) : \mathbf{u} \in \mathbb{Z}^m \text{ and } U = \langle \mathbf{u}, \mathbf{G} \rangle\}. \quad (27)$$

Protocol B.1: PoKE

Common input: A class group description $\mathbf{gk}$, public bases $\mathbf{G} = (G_1, \dots, G_m)$, and $U \in \mathbb{G}$.

Prover input: $\mathbf{u} \in \mathbb{Z}^m$ such that $U = \langle \mathbf{u}, \mathbf{G} \rangle$.

1. The verifier samples $\ell \leftarrow \text{Primes}(2\lambda)$ and sends ℓ to the prover.
2. For every $j \in [m]$, the prover computes the canonical remainder $r_j \in [0, \ell)$ of u_j modulo ℓ and sets $q_j = (u_j - r_j)/\ell$. It sends

$$Q := \langle \mathbf{q}, \mathbf{G} \rangle \quad \text{and} \quad \mathbf{r} = (r_1, \dots, r_m).$$

3. The verifier accepts if every r_j lies in $[0, \ell)$ and

$$U = \ell \cdot Q + \langle \mathbf{r}, \mathbf{G} \rangle. \quad (28)$$

Completeness follows from $\mathbf{u} = \ell \mathbf{q} + \mathbf{r}$. Integer extraction, however, fails in any group with efficient halving. For one base G , set $U := \text{Half}(G)$. On every odd challenge ℓ , the responses

$$Q := -U, \quad r := \frac{\ell + 1}{2} \in [0, \ell),$$

are accepted, since

$$\ell \cdot Q + r \cdot G = -\ell \cdot U + (\ell + 1) \cdot U = U.$$

For challenges with product L , the integer CRT lift is $z = (L + 1)/2$, while $z \cdot G = U$ would imply $L \cdot G = 0$. Thus, except in the low-order event, MCA's integer candidate does not open U : the transcripts encode $1/2$. We therefore use the dyadic relation $\mathcal{R}_{\text{PoKE}}^{\mathbb{Z}[1/2]}$ of Equation (4). For odd ℓ every $u_j \in \mathbb{Z}[1/2]$ still has a canonical residue $r_j \in [0, \ell)$, namely $a(2^\ell)^{-1} \bmod \ell$ for $u_j = a/2^\ell$, and then $q_j = (u_j - r_j)/\ell \in \mathbb{Z}[1/2]$. [Protocol B.1](#) therefore runs unchanged on a witness in $\mathbb{Z}[1/2]^m$.

Theorem B.1 (Dyadic PoKE). *Under the dyadic modular consistency assumption of Definition 4.2, [Protocol B.1](#) is a public-coin argument of knowledge for $\mathcal{R}_{\text{PoKE}}^{\mathbb{Z}[1/2]}$.*

The extractor rewinds the prover to collect accepting transcripts for s distinct prime challenges, with s large enough that their product exceeds $2B_T 2^{d_T}$. It then enumerates the candidate set $\mathcal{X}_{d_T, \ell}$ of Equation (3) and outputs the candidate that opens U . Finding no such candidate is exactly the DMC winning condition, so the extractor fails only with negligible probability.

Proof. Completeness is immediate from $\mathbf{u} = \ell \mathbf{q} + \mathbf{r}$ in $\mathbb{Z}[1/2]^m$.

For knowledge soundness, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an expected polynomial-time adversary with success probability $\epsilon(\lambda)$. Let T bound the running time of $\mathcal{A}_1$, and choose $s = \text{poly}(\lambda)$ such that

$$2^{(2\lambda-1)s} > 2B_T 2^{d_T}. \quad (29)$$

Transcript extractor. Define $\text{TrExt}^{A_2}(\text{st}, s)$ as follows. After $\mathcal{A}_1$ outputs (U, st) , rewind $\mathcal{A}_2(\text{st}, \cdot)$ and repeatedly sample $\ell \leftarrow \text{Primes}(2\lambda)$. Store $(\mathbf{r}, Q, \ell)$ whenever ℓ is fresh and

$$U = \langle \mathbf{r}, \mathbf{G} \rangle + \ell \cdot Q. \quad (30)$$

Stop once

$$\text{Tr} = [(\mathbf{r}_i, Q_i, \ell_i)]_{i=1}^s$$

has been obtained. Its expected running time is $O(s/\epsilon(\lambda))$; challenge collisions contribute a negligible term.

Witness extractor. Let $\ell = (\ell_1, \dots, \ell_s)$ and enumerate

$$\mathcal{X} := \mathcal{X}_{d_T, \ell}((\mathbf{r}_i)_{i=1}^s).$$

For each $\mathbf{x} \in \mathcal{X}$, test $U = \langle \mathbf{x}, \mathbf{G} \rangle$. Output the first such $\mathbf{x}$, and output $\perp$ if none exists. Since $|\mathcal{X}| \leq d_T + 2 = \text{poly}(\lambda)$, this is expected polynomial time.

Reduction to DMC. Suppose the witness extractor outputs $\perp$ with non-negligible probability. Define $(\mathcal{B}_1, \mathcal{B}_2)$ by

$$\begin{aligned} \mathcal{B}_1(\text{gk}, \mathbf{G}) : (U, \text{st}) &\leftarrow \mathcal{A}_1(\text{gk}, \mathbf{G}); \quad \text{return } (U, \text{st}), \\ \mathcal{B}_2(\text{st}) : \text{Tr} &\leftarrow \text{TrExt}^{A_2}(\text{st}, s); \quad \text{return Tr.} \end{aligned}$$

Equation (30) gives the required transcript equalities. Moreover, every ℓ_i has 2λ bits, so Equation (29) gives $L := \prod_i \ell_i > 2B_T 2^{d_T}$. Finally, output $\perp$ means

$$\forall \mathbf{x} \in \mathcal{X}_{d_T, \ell}((\mathbf{r}_i)_i), \quad U \neq \langle \mathbf{x}, \mathbf{G} \rangle,$$

which is precisely the DMC winning condition. Hence

$$\Pr[\text{Ext}^A = \perp] \leq \text{Adv}_{B,T,m,s}^{\text{DMC}}(\lambda) + \text{negl}(\lambda) = \text{negl}(\lambda).$$

Therefore the extractor outputs a witness in $\mathcal{R}_{\text{PoKE}}^{\mathbb{Z}[1/2]}$ except with negligible probability. $\square$

B.2 The batched SPAcc reduction

The reduction summarized in Section 6.2, in full.

Protocol B.2: Batched reduction $\Pi_{\text{MultiSPAcc}}^{(d)}$

Preprocessing input: A class group description gk and one fixed family $\mathbf{X} \in \mathcal{C}_{\Delta}^n$.

1. Compute and validate $(F_i)_i, T, B$ once, exactly as in Protocol 6.1.
2. Derive $A_T, A_B, \mathbf{H}_1, \dots, \mathbf{H}_d$ independently, compute C_{fix} , and publish one shared prover and verifier key containing this data.

Common input: The shared verification key and targets $\mathbf{Y} = (Y_1, \dots, Y_d) \in \mathcal{C}_{\Delta}^d$.

Prover input: The shared prover key and subsets $I_1, \dots, I_d \subseteq [n]$ satisfying Equation (16).

1. For each $h \in [d]$, compute the witness $\mathbf{w}_h = (P_h, \bar{P}_h, r_h, s_h, t_h, u_h, \kappa_h)$ exactly as in Protocol 6.1 for (Y_h, I_h) .
2. Send the single online commitment

$$U := \sum_{h=1}^d \langle \mathbf{w}_h, \mathbf{H}_h \rangle.$$

3. Output the two-block PoKEMath instance with commitments (C_{fix}, U) , opening blocks (T, B) and $(\mathbf{w}_1, \dots, \mathbf{w}_d)$, and predicate $\Phi_{\mathbf{Y}}^{(d)} = 0$.

B.3 Finding prime-leading representatives

We now take a small detour to explain the preprocessing costs of Protocol 6.1, as it is the most expensive part of our protocol, although it only has to be done once and can be updated if instance sizes need to be increased. Preprocessing needs, for every class X_i , a properly equivalent form whose leading coefficient is a distinct odd prime not dividing Δ . This also fixes the bit length $\rho(\lambda)$ of a preprocessed prime, which the range costs of Section 6.3 are stated in terms of.

A form represents an integer precisely when it is properly equivalent to a form with that leading coefficient: the leading coefficient of X^M is $X(r, t)$, where $(r, t)^\top$ is the first column of M . Any coprime pair (r, t) extends by Bézout to a matrix $M = \begin{pmatrix} r & s \\ t & u \end{pmatrix} \in \text{SL}_2(\mathbb{Z})$, so it suffices to search coprime pairs until the value $X(r, t)$ is prime. This leads to the following search algorithm.

Algorithm B.1: PrimeRep($X, \mathcal{P}$)

Input: A reduced form $X = (a, b, c)$ of discriminant Δ and the set $\mathcal{P}$ of primes already assigned to other indices.

Output: A prime form $F = (p, \beta, \gamma)$ and $M \in \text{SL}_2(\mathbb{Z})$ with $F = X^M$, where $p \notin \mathcal{P}$ is an odd prime and $p \nmid \Delta$.

For each coprime pair (r, t) with $t > 0$ or $(r, t) = (1, 0)$, in order of increasing $H := \max\{|r|, |t|\}$:

1. Set $p \leftarrow X(r, t) = ar^2 + brt + ct^2$. Continue to the next pair if $p \in \mathcal{P}$, if $p = p_\Delta$, or if p fails a primality test.
2. Choose $s, u \in \mathbb{Z}$ with $ru - st = 1$, set $M \leftarrow \begin{pmatrix} r & s \\ t & u \end{pmatrix}$, and return (X^M, M) , evaluated with Equation (1).

Since $\gcd(r, t) = 1$, Bézout supplies s, u with $ru - st = 1$, so $F = X^M$ is a primitive positive-definite form of discriminant Δ properly equivalent to X , with leading coefficient $X(r, t) = p$ and, by matching discriminants, $\gamma = (\beta^2 - \Delta)/(4p)$; thus $\beta^2 \equiv \Delta \pmod{4p}$ and F is a prime form in the sense of Equation (2).

Oddness is automatic, and it is needed: **ComposeCoprime** takes odd leading coefficients, and Equation (9) determines B modulo $2T$. As in the proof of Lemma 6.1, $\Delta \equiv 5 \pmod{8}$ forces a, b, c odd, so $X \equiv X_1^2 + X_1X_2 + X_2^2 \pmod{2}$ is anisotropic over $\mathbb{F}_2$ and a coprime pair is nonzero modulo 2 and so no value of X is ever even. Finally, M is a certificate of proper equivalence, so the one-time validator can check $\det M = 1$ and recompute X^M instead of reducing F .

The size of the prime is unconditional too: $p \leq (|a| + |b| + |c|)H^2 = O(|\Delta|H^2)$, since reduction gives $a \leq \sqrt{|\Delta|/3}$, $|b| \leq a$ and $c \leq |\Delta|/4 + O(\sqrt{|\Delta|})$. Over the range of H used below this gives $\rho(\lambda) = |\Delta|_{\text{bit}} + O(\log|\Delta|_{\text{bit}})$, which is the bound on the bit length of a preprocessed prime quoted in the costs above.

The number of candidates examined before a prime appears is only conjectured. We heuristically assume that the values $X(r, t)$ contain primes at the density predicted for integers of their size, so that a candidate is prime with probability $\Omega(1/\log p)$. (No prime divides every value and makes the search run forever: if q did, then $q \mid X(1, 0) = a$, $q \mid X(0, 1) = c$ and $q \mid X(1, 1) = a + b + c$, hence $q \mid \gcd(a, b, c) = 1$.)

Under that assumption the cost follows. The coprime pairs with $\max\{|r|, |t|\} \leq H$ number $\Theta(H^2)$, so a box of size $H = O(\sqrt{|\Delta|_{\text{bit}}})$ already supplies the $O(|\Delta|_{\text{bit}})$ candidates expected before a prime appears, and each candidate costs one evaluation of X together with a constant number of Miller–Rabin rounds. The expected cost is $O(|\Delta|_{\text{bit}}^2 M(|\Delta|_{\text{bit}}))$ bit operations per index, and in practice trial division by small primes discards most candidates before any Miller–Rabin test. This one-time preprocessing for the fixed family $\mathbf{X}$ is $\text{poly}(\lambda)$ per index, independent of the online target.

Remark B.2 (Evidence for the conjectured prime density). We ran [Algorithm B.1](#) in SageMath using the parameters of Section 4.1: taking a mean of hundred different 3072-bit prime discriminants, and for each of them a base class G together with ten classes $X_i = 2^{i-1} \cdot G$. The search examined an average of 749 candidate pairs per class, with standard deviation 930 and median 419; trial division by the primes below 10^4 left 92 primality tests per class, and a class took 0.28 seconds. The primes found averaged 2135 bits, inside $\rho(\lambda)$.

A uniformly random *odd* integer of that size is prime with probability $2/\log p$ and would need $\log p/2 \approx 740$ trials, against the 749 measured. To the precision of this experiment the values of X are almost exactly as likely to be prime as random integers of their size. The per-discriminant means ranged from 150 to 2368 with standard deviation 424 around the common mean.

C Deferred Proofs

This section contains the deferred proofs from Sections 5, 6 and 7. The proof of Theorem B.1 appears with its protocol in Appendix B.

The following lemma identifies the response in a fresh accepting transcript with the reduction of the extracted PoKE witness. It is the step that ties the predicate check of BatchPoKEMath to the extracted opening.

Lemma C.1 (Remainder consistency). *Assume DMC and DLOG, and let $\mathbf{u} \leftarrow \text{Ext}_{\text{PoKE}}^A$ be the witness extracted from a stateful prover, and let $(\ell, Q, \mathbf{r})$ be a fresh accepting PoKE transcript. Then*

$$\mathbf{r} = \mathbf{u} \pmod{\ell}$$

except with negligible probability.

Proof. Suppose the contrary event has non-negligible probability. Rewind the prover and collect s accepting inconsistent transcripts $(\ell_i, Q_i, \mathbf{r}_i)$ for distinct challenges, where s satisfies Equation (29). Set

$$\begin{aligned} \rho_i &:= \mathbf{u} \bmod \ell_i, & \mathbf{q}_i &:= \frac{\mathbf{u} - \rho_i}{\ell_i}, \\ \mathbf{a}_i &:= (\mathbf{r}_i - \rho_i) \bmod \ell_i, & \mathbf{b}_i &:= \frac{\mathbf{r}_i - \rho_i - \mathbf{a}_i}{\ell_i}. \end{aligned}$$

Subtracting the two openings of U gives

$$0 = \langle \mathbf{a}_i, \mathbf{G} \rangle + \ell_i \cdot (Q_i - \langle \mathbf{q}_i, \mathbf{G} \rangle + \langle \mathbf{b}_i, \mathbf{G} \rangle). \quad (31)$$

Thus Equation (31) gives s valid DMC transcripts for the commitment 0. DMC yields some

$$\mathbf{x} \in \mathcal{X}_{dT, \ell}((\mathbf{a}_i)_i) \quad \text{such that} \quad \langle \mathbf{x}, \mathbf{G} \rangle = 0.$$

Every candidate in this set is congruent to $\mathbf{a}_i$ modulo ℓ_i . Since every collected transcript is inconsistent, $\mathbf{a}_i \neq \mathbf{0}$, and hence $\mathbf{x} \neq \mathbf{0}$. Clearing the power-of-two denominator of $\mathbf{x}$ gives a nonzero integer vector $\hat{\mathbf{x}}$ with $\langle \hat{\mathbf{x}}, \mathbf{G} \rangle = 0$, contradicting the discrete-log assumption. $\square$

Theorem 5.1 (Batched PoKEMath). *Assume DMC and the discrete-log assumption. Then Protocol 5.1 is a public-coin argument of knowledge for $\mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], k}$ for polynomial k . With L_q and C_f as in the theorem statement, the prover performs $\text{MSM}(m, L_q + 2\lambda)$ group operations and $O(mM(L_q))$ bit operations; the verifier performs one $(k + m + 1)$ -term MSM whose scalars have at most $4\lambda + 1$ bits and $O((C_f + m)M(2\lambda))$ bit operations; and the proof is one group element and m elements of $\mathbb{F}_\ell$.*

Proof. Completeness follows by taking the indicated linear combination of the k opening equations, and by reducing $f(\mathbf{u}_1, \dots, \mathbf{u}_k) = 0$ modulo the odd prime ℓ .

For knowledge soundness, first fix the prime challenge ℓ and the response vectors $(\mathbf{r}_j)_j$, and define

$$D_j := U_j - \langle \mathbf{r}_j, \mathbf{G}_j \rangle.$$

Rewind after this prefix to obtain k accepting continuations with distinct batching challenges $\eta_1, \dots, \eta_k$ and quotient elements $Q_1, \dots, Q_k$. This is the usual heavy-row/generalized-forking step; because k is polynomial and $|\mathbb{F}_\ell|$ is exponential, it succeeds in expected polynomial time whenever the original success probability is non-negligible.

For the two-block case, two continuations with $\eta \neq \eta'$ give

$$D_1 + \eta D_2 = \ell Q_\eta, \quad D_1 + \eta' D_2 = \ell Q_{\eta'}.$$

Subtracting shows $(\eta - \eta')D_2 = \ell(Q_\eta - Q_{\eta'})$. Since $\gcd(\eta - \eta', \ell) = 1$, Bézout coefficients turn the right-hand side into an explicit ℓ -th root of D_2 , after which either equation gives one for D_1 . The general case is the same linear-algebra argument.

Let $M \in \mathbb{Z}^{k \times k}$ have entry $M_{t,j}$ equal to the canonical representative of $\eta_t^{j-1} \in \mathbb{F}_\ell$. The accepting equations are

$$\sum_{j=1}^k M_{t,j} D_j = \ell Q_t \quad (t \in [k]). \quad (32)$$

The reduction of M modulo ℓ is a Vandermonde matrix on distinct points, so $\delta := \det(M)$ is coprime to ℓ . Multiplying Equation (32) by $\text{adj}(M)$ gives, for each j , an efficiently computable $R_j \in \mathbb{G}$ such that

$$\delta D_j = \ell R_j.$$

Choose $x, y \in \mathbb{Z}$ with $x\delta + y\ell = 1$ and set $\widehat{Q}_j := xR_j + yD_j$. Then

$$\ell \widehat{Q}_j = x\ell R_j + y\ell D_j = (x\delta + y\ell)D_j = D_j.$$

Thus the batched continuations yield an individual accepting PoKE equation

$$U_j = \ell \widehat{Q}_j + \langle \mathbf{r}_j, \mathbf{G}_j \rangle$$

for every block, even though the prover transmitted only one quotient in each continuation.

Repeat this inner extraction for the prime challenges required by the dyadic PoKE extractor. Applying that extractor of Theorem B.1 and Lemma C.1 blockwise gives $\mathbf{u}_j \in \mathbb{Z}[1/2]^{m_j}$ with $U_j = \langle \mathbf{u}_j, \mathbf{G}_j \rangle$ and $\mathbf{r}_j = \mathbf{u}_j \bmod \ell$ for every j .

It remains to prove that the joint predicate vanishes over $\mathbb{Z}[1/2]$. Write $\phi := f(\mathbf{u}_1, \dots, \mathbf{u}_k)$ and assume $\phi \neq 0$ with non-negligible probability. We construct an adaptive-root adversary $(\mathcal{B}_1, \mathcal{B}_2)$. In its first stage, $\mathcal{B}_1$ runs the prover together with the blockwise extractor, samples an independent $H \in \mathbb{G}$, and outputs $V := \phi \cdot H$. After receiving the adaptive-root challenge ℓ , adversary $\mathcal{B}_2$ forwards ℓ to the prover and obtains an accepting response. That response carries only the batched quotient, so $\mathcal{B}_2$ first repeats the η -forking above under this fixed ℓ , recovering per-block quotients $\widehat{Q}_j$ with $U_j = \ell \widehat{Q}_j + \langle \mathbf{r}_j, \mathbf{G}_j \rangle$; each block then carries an individual accepting PoKE transcript, which is what the next step needs. By Lemma C.1, $\mathbf{r}_j = \mathbf{u}_j \bmod \ell$ for every j , so, since f has integer coefficients, Equation (7) gives

$$\phi = f(\mathbf{r}_1, \dots, \mathbf{r}_k) = 0 \pmod{\ell}.$$

Consequently ϕ/ℓ lies in $\mathbb{Z}[1/2]$, and $\mathcal{B}_2$ outputs $X := (\phi/\ell) \cdot H$, for which $\ell \cdot X = \phi \cdot H = V$. Moreover $V \neq 0$ except with negligible probability: a uniformly sampled H is nonzero except with negligible probability, and, after clearing the power-of-two denominator of ϕ , the event $V = 0$ for nonzero H violates the low-order property. Lemmas A.4 and A.5 derive both adaptive-root and low-order security from DMC, so a nonzero ϕ contradicts DMC. Hence $f(\mathbf{u}_1, \dots, \mathbf{u}_k) = 0$ over $\mathbb{Z}[1/2]$ except with negligible probability, and the extracted vectors form a witness for $\mathcal{R}_{\text{PoKE Math}}^{\mathbb{Z}[1/2], k}$.

For the costs, reducing each witness coordinate modulo ℓ and dividing the difference by ℓ costs $O(M(L_q))$ bit operations per coordinate, and the quotient coordinates have bit length at most L_q . Forming Q is one m -term MSM whose scalars are products of a batching weight with a quotient

coordinate, hence of at most $L_q + 2\lambda$ bits. The prover sends one quotient element and all m response coordinates. Equation (6) contains k commitment terms, one quotient term, and m response terms. Products of a batching weight with a response have at most $4\lambda + 1$ bits, giving the stated verifier cost, and evaluating f once while processing the m responses gives the stated bit cost. $\square$

Lemma 6.1 (Dyadic rigidity of integral forms). *Let $\Delta < 0$ satisfy $\Delta \equiv 5 \pmod{8}$, and let f and g be primitive positive-definite integral forms of discriminant Δ . If*

$$g = f^M \quad \text{for some } M \in \text{SL}_2(\mathbb{Z}[1/2]),$$

then $M \in \text{SL}_2(\mathbb{Z})$. Hence f and g are properly equivalent integral forms.

Proof. Write $f = (a, b, c)$. Since $b^2 - 4ac = \Delta \equiv 5 \pmod{8}$, the integers a, b, c are odd: first b is odd, and then $4ac = b^2 - \Delta \equiv 4 \pmod{8}$, so ac is odd. Hence

$$f(X, Y) \equiv X^2 + XY + Y^2 \pmod{2}.$$

The three nonzero pairs in $\mathbb{F}_2^2$ all evaluate to 1, so this form is anisotropic. Write v_2 for the 2-adic valuation, with $v_2(0) = +\infty$. For nonzero $(x, y) \in \mathbb{Z}[1/2]^2$,

$$v_2(f(x, y)) = 2 \min\{v_2(x), v_2(y)\}. \quad (33)$$

Indeed, put $k = \min\{v_2(x), v_2(y)\}$ and write $(x, y) = 2^k(x_0, y_0)$, where $x_0, y_0 \in \mathbb{Z}$ and at least one is odd. Anisotropy gives $f(x_0, y_0) \equiv 1 \pmod{2}$, while homogeneity gives $f(x, y) = 2^{2k}f(x_0, y_0)$, proving the equality.

Let the columns of M be $(r, t)^\top$ and $(s, u)^\top$. The leading and trailing coefficients of $g = f^M$ are $f(r, t)$ and $f(s, u)$. They are nonzero because M is invertible and f is positive definite, and they are integers because g is integral. Their 2-adic valuations are therefore nonnegative. Equation (33) applied to each column gives

$$\min\{v_2(r), v_2(t)\} \geq 0, \quad \min\{v_2(s), v_2(u)\} \geq 0.$$

Thus every entry of M lies in $\mathbb{Z}$. Since $\det M = 1$, we obtain $M \in \text{SL}_2(\mathbb{Z})$, as claimed. $\square$

Theorem 6.2 (Class-group SPAcc reduction). *Assume the discrete-log assumption of Definition 4.1. For every fixed family $\mathbf{X} \in \mathcal{C}_\Delta^n$, Protocol 6.1 is a reduction of knowledge*

$$\mathcal{R}_{\text{SPAcc}}^{\text{gk}, \mathbf{X}} \longrightarrow \mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 2}.$$

Let L bound the bit length of Δ , T , B , the coefficients of Y , and every coordinate of the target witness $(T, B, \mathbf{w})$, and put $D := |\Delta|_{\text{bit}}$. The reduction costs are:

- Preprocessing: n searches for prime representatives, n validation checks, $\text{MSM}(2, L)$ group operations, and $O(M(L) \log n)$ bit operations.
- Prover: $\text{MSM}(7, L)$ group operations and $O(M(L)(\log n + \log L))$ bit operations.
- Verifier: no group operations and $O(D)$ bit operations.
- Communication: one group element.

Proof. For completeness, let $I \subseteq [n]$ and set $P = P_I$ and $\bar{P} = T/P$. By coprime composition, $F(P, B)$ represents $\sum_{i \in I} X_i = Y$. Hence there is an $M \in \text{SL}_2(\mathbb{Z})$ with $Y^M = F(P, B)$. Taking $\kappa = 0$ gives

both opening equations and $\Phi_Y = 0$. The target instance is a deterministic function of Y and the transmitted element U , so the reduction is public.

For knowledge soundness, suppose the target witness $(\tau, \beta, P, \bar{P}, r, s, t, u, \kappa) \in \mathbb{Z}[1/2]^9$ makes the target instance valid. The fixed commitment then has openings (T, B) and (τ, β) . The discrete-log assumption, after clearing powers of two, gives

$$\tau = T, \quad \beta = B. \quad (34)$$

Because Φ_Y is a sum of squares over the ordered ring $\mathbb{Z}[1/2]$, all four constituent polynomials vanish. In particular,

$$P\bar{P} = T, \quad ru - st = 1, \quad (35)$$

and Y^M has leading coefficient P and middle coefficient $B + 2\kappa P$ for $M = \begin{pmatrix} r & s \\ t & u \end{pmatrix}$.

The leading-coefficient equation gives $P = Y(r, t) > 0$, since the first column of M is nonzero and Y is positive definite. For every odd prime, a dyadic rational has nonnegative valuation. Since $T = \prod_i p_i$ is square-free, Equation (35) therefore determines a unique subset

$$I := \{i : v_{p_i}(P) = 1\} \quad \text{and} \quad P = 2^d P_I$$

for some $d \in \mathbb{Z}$. Equation (33), applied with $f = Y$ to the first column of M , shows that $d = 2e$ is even.

Set

$$D_e := \begin{pmatrix} 2^{-e} & 0 \\ 0 & 2^e \end{pmatrix}, \quad S := \begin{pmatrix} 1 & -\kappa 2^{2e} \\ 0 & 1 \end{pmatrix}.$$

Both matrices lie in $\text{SL}_2(\mathbb{Z}[1/2])$. Right multiplication by D_e changes the leading coefficient of Y^M to P_I and leaves its middle coefficient unchanged; right multiplication by S changes that middle coefficient to B . The discriminant fixes the final coefficient, so

$$Y^{MD_e S} = F(P_I, B).$$

$F(P_I, B)$ is integral by construction. It is primitive because a prime dividing all three coefficients would divide both P_I and B , and hence Δ ; this is impossible since every $p_i \nmid \Delta$. Thus both sides are primitive integral forms. Lemma 6.1 implies $MD_e S \in \text{SL}_2(\mathbb{Z})$. Hence Y is properly equivalent to $F(P_I, B)$, whose class is $\sum_{i \in I} X_i$. The extracted subset I is therefore a valid source witness.

For the costs, a balanced tree has $\lceil \log n \rceil$ levels, and the total bit length of the operands at any level is $O(L)$. Superlinearity of M therefore bounds each level by $O(M(L))$, so the tree computes T in $O(M(L) \log n)$ bit operations. A subproduct tree followed by the usual upward CRT combination computes B within the same bound. The fixed commitment $C_{\text{fix}} = TA_T + BA_B$ is one two-term MSM; prime search and validation contribute the separately listed n calls each.

Online, product trees compute P and $\bar{P}$ in $O(M(L) \log n)$ bit operations. Tracking the change-of-variables matrix through composition and reduction costs $O(M(L) \log L)$ further bit operations. Forming U is one seven-term MSM with L -bit scalars. The verifier only copies the D -bit coefficients of Y into the target predicate, and the sole message is U . $\square$

Theorem 6.4 (Batched SPAcc).

Assume the discrete-log assumption of Definition 4.1. For every family $\mathbf{X} \in \mathcal{C}_\Delta^n$ and polynomial d , Protocol B.2 is a reduction of knowledge

$$\left(\mathcal{R}_{\text{SPAcc}}^{\text{gk}, \mathbf{X}} \right)^d \longrightarrow \mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 2}.$$

It performs the accumulator preprocessing once, sends one online group element, and produces PoKEMath blocks of dimensions 2 and $7d$. Under DMC and the discrete-log assumption, composition with [Protocol 5.1](#) sends two group elements and $2 + 7d$ field elements.

Proof. The preprocessed values $(F_i)_i, T, B$ depend only on $\mathbf{X}$, so the same validated record is valid for every target. For completeness, apply the witness construction of [Protocol 6.1](#) independently to each (Y_h, I_h) . Every summand in Equation (17) vanishes, and the two advertised commitments have the stated openings.

For knowledge soundness, extract (τ, β) from the fixed block and $\mathbf{w}_1, \dots, \mathbf{w}_d$ from the online block. The discrete-log assumption identifies $(\tau, \beta) = (T, B)$ exactly as in Equation (34). Since $\mathbb{Z}[1/2]$ is ordered and $\Phi_{\mathbf{Y}}^{(d)}$ is a sum of squares, every constituent $\Phi_{Y_h}(T, B, \mathbf{w}_h)$ vanishes. Apply the extraction argument of Theorem 6.2 separately to each h to recover a subset $I_h \subseteq [n]$ satisfying $Y_h = \sum_{i \in I_h} X_i$. These subsets form a witness for the product relation.

Only the seven-coordinate online blocks are repeated: their independent bases let the prover combine them into one commitment U . The fixed block remains two-dimensional, proving the reduction costs. Finally, [Protocol 5.1](#) contributes one quotient element for both opening blocks and $2 + 7d$ responses, giving the stated composed communication. $\square$

Corollary 6.3 (SPAcc argument). *Assume DMC and the discrete-log assumption of Section 4.3. The sequential composition of Π_{SPAcc} with the two-block form of PoKEMath is a public-coin argument of knowledge for $\mathcal{R}_{\text{SPAcc}}^{\text{gk}, \mathbf{X}}$.*

Proof. Apply [Protocol 5.1](#) with $k = 2$ to the fixed and online opening blocks. Its extractor handles the two opening equations separately before applying the PoKEMath argument to their joint predicate. Composing this argument with Theorem 6.2 and applying Theorem A.1 proves the claim. $\square$

Corollary 6.5 (Batched class group range argument). *For independently sampled bases $\mathbf{G}$, under DMC and the discrete-log assumption, [Protocol 6.2](#) is a public-coin argument of knowledge for the batched relation $\mathcal{R}_{\text{Range}}^{\mathbf{a}, \mathbf{b}}$ of Section 3.4. It is not zero knowledge.*

Let $\rho(\lambda)$ bound the bit length of each preprocessed prime representative; by Section 6.1.2, $\rho(\lambda) = |\Delta|_{\text{bit}} + O(\log|\Delta|_{\text{bit}})$. Let $s(n, \lambda)$ be the bit-length bound supplied by the witness-construction algorithm for the honest accumulator, equivalence, and PoKEMath-quotient coordinates, where $s(n, \lambda) = O(n\rho(\lambda) + |\Delta|_{\text{bit}} + \lambda)$. Put $A := \max_j \max\{|a_j|_{\text{bit}}, |b_j|_{\text{bit}}\}$. For $n = \sum_j n_j$, the costs are:

- Preprocessing: n prime-representative searches, n validation checks, $n - m$ doublings,

$$\text{MSM}(2, n\rho(\lambda) + 1) + 2 \text{MSM}(m, A)$$

group operations, and $O(M(n\rho(\lambda)) \log n)$ bit operations. The verification key contains $m + 19$ group elements; the prover key additionally contains n prime forms and T, B .

- Prover: two group additions, one inversion,

$$\text{MSM}(14, s(n, \lambda)) + \text{MSM}(16, s(n, \lambda) + 2\lambda)$$

group operations, and

$$O(M(s(n, \lambda))(\log(n + 1) + \log s(n, \lambda)))$$

bit operations.

- Verifier: two group additions and one inversion; group cost $\text{MSM}(19, 4\lambda+1)$ and bit cost $O(M(|\Delta|_{\text{bit}}) + 58M(2\lambda))$.
- Proof size: two group elements and 16 elements of $\mathbb{F}_\ell$, or at most 32λ bits outside the group elements, independently of m .

For fixed λ , $s(n, \lambda) = O(n)$; thus the prover performs a constant number of MSMs with $O(n)$ -bit scalars and $O(M(n) \log n)$ bit operations. The case $m = 1$ is a single interval.

Proof. If $a_j \leq v_j \leq b_j$ for every j , then $v_j - a_j$ and $b_j - v_j$ lie in $[0, b_j - a_j] \subseteq [0, 2^{n_j})$, so their binary expansions select a subset of the block of $\mathbf{X}$ belonging to G_j ; the union over j is one subset of $\mathbf{X}$ for each sign. Completeness then follows from Theorem 6.2 and PoKEMath completeness.

For soundness, extract the PoKEMath witness. Since $\Psi_{V, \mathbf{a}, \mathbf{b}} = 0$ is a sum of squares over $\mathbb{Z}[1/2]$, both SPAcc predicates vanish. Repeating the extraction argument of Theorem 6.2 yields subsets $I_-, I_+ \subseteq [n]$. Partitioning each by the base its elements belong to gives $\mathbf{z}_-, \mathbf{z}_+ \in \mathbb{Z}^m$ with $0 \leq z_{\pm, j} < 2^{n_j}$ and

$$V - \langle \mathbf{a}, \mathbf{G} \rangle = \langle \mathbf{z}_-, \mathbf{G} \rangle, \quad \langle \mathbf{b}, \mathbf{G} \rangle - V = \langle \mathbf{z}_+, \mathbf{G} \rangle.$$

Adding gives $\langle \mathbf{b} - \mathbf{a}, \mathbf{G} \rangle = \langle \mathbf{z}_- + \mathbf{z}_+, \mathbf{G} \rangle$. All exponents are polynomially bounded, so the multi-generator discrete-log assumption implies $\mathbf{z}_- + \mathbf{z}_+ = \mathbf{b} - \mathbf{a}$ coordinatewise. Thus $\mathbf{v} := \mathbf{a} + \mathbf{z}_- = \mathbf{b} - \mathbf{z}_+$ satisfies $a_j \leq v_j \leq b_j$ for every j and $V = \langle \mathbf{v}, \mathbf{G} \rangle$, as required.

For preprocessing, the power family costs $n_j - 1$ doublings in the block of G_j , hence $n - m$ in all, while the product and CRT trees cost $O(M(n\rho(\lambda)) \log n)$ bit operations. The fixed commitment is a two-term MSM with $(n\rho(\lambda) + 1)$ -bit scalars; the two endpoint combinations $A_\pm$ are m -term MSMs with A -bit scalars.

Online, the prover first forms Y_-, Y_+ from the preprocessed endpoint offsets with two additions and one inversion. One fourteen-term MSM then commits both SPAcc witnesses. Batched PoKEMath forms its single quotient with one sixteen-term MSM over the fixed and online blocks. Product, equivalence, remainder, and quotient computations account for the stated bit cost in $s(n, \lambda)$.

The verifier forms Y_-, Y_+ from the same offsets using two additions and one inversion, then checks both openings with one nineteen-term batched MSM. One copy of Φ uses 1 multiplication for h_{prod} , 2 for h_{det} , 6 for h_{lead} , 8 for h_{mid} , and 4 to square these values, hence 21 in total. The two copies use 42 field multiplications; processing the 16 responses explains the factor 58, while reducing the form coefficients contributes $O(M(|\Delta|_{\text{bit}}))$. The proof is U , one quotient commitment, and $2 + 14$ field responses: two group elements and sixteen field elements. $\square$

Lemma 7.1 (Binding of the integer-encoding commitment). *Under the discrete-log assumption of Definition 4.1, CS_{Mist} is binding.*

Proof. Suppose an adversary produces two accepting openings (p, d) and (p', d') of the same commitment C , with $p \neq p'$. Put $\delta_i := d_i - d'_i$. Since the two coefficient vectors differ modulo q , the vector δ is nonzero. By Equation (22),

$$e \cdot G = 0, \quad e := \sum_{i=0}^{N-1} \delta_i \alpha^i \in \mathbb{Z}[1/2]. \quad (36)$$

We first show that $e \neq 0$. If $e = 0$, let j be the largest index for which $\delta_j \neq 0$. Two distinct elements of $\mathcal{H}$ have denominators at most B , so their difference has denominator at most B^2 and a

nonzero integer numerator; hence $|\delta_j| \geq 1/B^2$, while the magnitude bound gives $|\delta_i| \leq 2A$ for every i . Consequently,

$$\frac{\alpha^j}{B^2} \leq |\delta_j| \alpha^j = \left| \sum_{i=0}^{j-1} \delta_i \alpha^i \right| \leq 2A \sum_{i=0}^{j-1} \alpha^i < \frac{2A\alpha^j}{\alpha - 1}.$$

Because α is an integer and $\alpha > 2AB^2$, we have $\alpha - 1 \geq 2AB^2$, contradicting the displayed strict inequality.

Thus e is nonzero. Although the δ_i are general rationals, e is the difference of the two committed values, each of which lies in $\mathbb{Z}[1/2]$ by the opening relation, so $e \in \mathbb{Z}[1/2]$. Clearing a common power-of-two denominator in Equation (36) gives a nonzero polynomial-size integer $\hat{e}$ with $\hat{e} \cdot G = 0$. The adversary can therefore output the two distinct one-dimensional integer representations $(\hat{e})$ and (0) of the same group element, contradicting the discrete-log assumption. Hence two different messages cannot open the same commitment except with negligible probability. $\square$

Theorem 7.2 (Weak evaluation). *Under the discrete-log assumption, and for a uniformly sampled evaluation point $\mathbf{z} \leftarrow \mathbb{F}_q^\mu$, Protocol 7.1 is a weak reduction of knowledge*

$$\mathcal{R}_{\text{Eval}} \longrightarrow \mathcal{R}_{\text{PoKEMath}}^{\mathbb{Z}[1/2], 3} \times \mathcal{R}_{\text{Range}}^{\mathbf{0}, \mathbf{B}-1}.$$

Sketch. See [BMSS25] for details. For completeness, let $c = \text{Enc}_\alpha(p)$. In the product $c\sigma_{\mathbf{z}}$, the coefficient at position $N-1$ is

$$\sum_{i=0}^{N-1} \tilde{p}_i \tilde{\chi}_i(\mathbf{z}).$$

The choice of α prevents a carry into this position, so the value t returned by **CoeffSplit** is this integer inner product. Therefore $t \equiv p(\mathbf{z}) = y \pmod{q}$, and $w = (t - \tilde{y})/q$ is an integer. The coefficient estimates from the same convolution give $0 \leq s < S_\star$, $0 \leq t < T_\star$, and $0 \leq u < U_\star$; these are precisely the strict DewTwo thresholds. Rounding as in Equation (25) therefore gives $0 \leq (s, t, u) \leq \mathbf{B} - \mathbf{1}$. Both target relations are consequently satisfied. The target instances are deterministic functions of the source instance and the transmitted pair (W, V) , so the reduction is public.

For knowledge soundness, extract $(c, w, s, t, u) \in \mathbb{Z}[1/2]^5$ from the three-block PoKEMath instance and $(s', t', u') \in \mathbb{Z}^3$ from the range instance. Both triples open V with respect to the independently sampled bases (G_s, G_t, G_u) . After clearing a common power-of-two denominator, the event $(s, t, u) \neq (s', t', u')$ gives two distinct integer representations of V ; the discrete-log assumption therefore implies, except with negligible probability,

$$(s, t, u) = (s', t', u') \tag{37}$$

In particular, the three values obey the power-of-two bounds in Equation (25). Since $g_{\mathbf{z}, y}$ is a sum of squares over the ordered ring $\mathbb{Z}[1/2]$, its vanishing yields

$$t = \tilde{y} + qw, \quad \sigma_{\mathbf{z}} c = s + t\alpha^{N-1} + u\alpha^N.$$

It remains to recover the opening, which is DewTwo's coefficient-recovery argument [BMSS25] applied to exactly these two equations and the rounded bounds; we sketch its three stages and what our substitution changes. Only the bounds change: each B_X is less than twice $X_\star$, so the coefficient estimates gain a single bit, while the base separation supplied by α is untouched.

A bounded rational form for the base- α coefficients. Analysing the coefficients at positions $N - 2$ and $N - 1$ together with the carries from lower positions turns the two equations into one of the form $d(\mathbf{z}) \equiv s \pmod{\alpha}$, where d is the coefficient vector read as a multilinear polynomial and both sides are bounded. The inverse Lipschitz–Schwartz–Zippel lemma of Bünz and Fisch [BF23, Corollary 1] then gives, except with probability $2^{-\lambda} + \mu/q$ over the sampling of $\mathbf{z}$, a representation of each base- α coefficient of c as $(m_i\alpha + n_i)/k_i$ with m_i, n_i, k_i bounded. This is the only step that uses the independence of $\mathbf{z}$ from C , and hence the only source of weakness in the statement. Its hypothesis is $\log B \geq 8\mu^2 + \lambda \log_2(2\mu)$, which is exactly why B was set to $N^{8\mu}(2\mu)^\lambda$.

Splitting and regrouping. The extractor splits each coefficient into $\alpha m_i/k_i$ and n_i/k_i and regroupes the pieces by the power of α they carry. The numerator and denominator bounds place the result in $\mathcal{H}^N$, so it is an opening of C in the sense of Equation (22); binding (Lemma 7.1) makes it the only one. This yields $d \in \mathcal{H}^N$ with

$$c = \sum_{i=0}^{N-1} d_i \alpha^i, \quad p_i := d_i \bmod q.$$

Correctness of the evaluation. The extracted coefficients are bounded, so no term overflows into a neighbouring power of α and the recovered p satisfies $p(\mathbf{z}) = y$. Here the choice $\alpha = q^k$ is what makes the argument close: the checks guarantee the evaluation only modulo α , and $q \mid \alpha$ transfers that guarantee to the claimed value in $\mathbb{F}_q$. Since $C = c \cdot G$, the pair (p, d) is a valid source witness, which proves weak knowledge soundness. $\square$