Bulletproofs∗: Verifier-Efficient Arithmetic Circuit Proofs via Folding

Emanuele Scala[1] and Daniele Bartoli[2]

[1] emsczkp@proton.me
[2] Department of Mathematics and Computer Science, University of Perugia

Abstract. We present *Bulletproofs∗* (BP∗, BulletproofsStar), a folding scheme for arithmetic circuit proofs under standard assumptions and without preprocessing, i.e., for the arithmetic circuit satisfiability language of Bulletproofs (S&P 18), following the recipe of ProtoStar (ePrint 2023/620). To this end, we first adapt the algebraic verifiers of the arithmetic circuit proof of Bulletproofs to the algebraic form required by ProtoStar, and prove that the modified protocol remains secure. Then, we design the *Bulletproofs∗* folding scheme that is complete and knowledge-sound. Finally, we analyze the resulting verifier cost after the folding-to-IVC transformation. The result shows an asymptotic linear gain compared to repeated invocations of the monolithic Bulletproofs verifier.

Keywords: Folding Scheme · Bulletproofs · ProtoStar · Zero-Knowledge Proofs · Incremental Verifiable Computation · Recursive Proofs.

1 Introduction

Folding schemes are the building blocks of Incrementally Verifiable Computations (IVCs): proofs that attest to the correct execution of a computation together with the validity of a previous proof [8].

Prior to folding schemes, recursive proofs realize IVCs where the expensive work of the verifier circuit is treated as part of the statement, by taking as public inputs both the input and the prover's claimed output. This allows the verifier to merge two instances of the statement into a single instance, thus amortizing the cost of the expensive work across many composed proofs. Finally, only at the end of the recursion, the ultimate verifier executes the expensive check to validate the entire sequence of proofs. However, this approach still requires the verifier circuit to partially verify a proof at each recursion step [19].

With folding schemes one can check two instances in some relation by checking a single (folded) instance. Then, instead of accumulating expensive steps, an IVC based on folding schemes accumulates instances themselves. Concretely, at each incremental step, the IVC prover proves that it has executed the step computation and that it has folded its prior step into a running relaxed instance, while the verifier circuit verifies that the prover correctly folds the previous step. A folding scheme is non-trivial when the resulting verifier time and communication are lower than a verifier checking each original instance separately [19].

ProtoStar [11] allows to build secure and efficient folding schemes from interactive arguments whose verifier is of special form. ProtoStar recipe involves the following ingredients: i) use succinct homomorphic commitment scheme; ii) apply Fiat-Shamir transform to obtain a non-interactive argument of knowledge (NARK); iii) build an accumulation scheme, where the accumulator is the NARK proof, by sampling a random challenge and taking a random linear combination of the current accumulator with the new proof.

In ProtoStar, the accumulated instances are the "relaxed" verifier checks of the NARK augmented with error terms that arise at each accumulation step: given two NARK proofs, the prover folds them via a random linear combination and generates a list of d error terms (with d the degree of the NARK verifier) as an accumulation proof. The verifier only needs to check that the accumulated instances are consistent with the accumulation proof and the original instances being folded. After all accumulation steps, a decider performs a final check to verify the consistency of commitments in the accumulator and that the "relaxed" NARK verifier check still passes. An accepting decider implies that all accumulated NARKs were valid. Finally, a folding-to-IVC transformation can be done by invoking the compiler of [12] which yields a secure IVC scheme.

1.1 Contribution

The purpose of this paper is to improve the verification efficiency of Bulletproofs-based Zero-Knowledge Proofs (ZKPs). This is particularly beneficial for confidential transaction protocols that rely on such trustless ZKP primitives [1, 16, 9, 23]. In a previous work, Bulletproofs+ [14] (BP+), a weighed Inner-Product Argument (IPA) is used to reduce the proof size, while maintaining verifier time comparable to Bulletproofs. Bulletproofs++ [15] (BP++) builds on BP+ and introduces a weighed norm argument, improving verifier time by a logarithmic factor. Other works, such as the Compressed Σ-IPA/protocol [2, 24] yield a lighter version of the IPA with fewer cryptographic operations. However, these approaches still work for the monolithic Bulletproofs verification. An attempt to apply recursive proofs to Bulletproofs-style arguments appears in Halo [8], which achieves amortized verifier time through proof batching. However, Halo modifies the IPA relation, resulting in a new polynomial evaluation proof, and is tied to the Sonic [21] proof system. In contrast, we apply the generic ProtoStar folding recipe without modifying the original Bulletproofs relation. Moreover, we enable a folding scheme for the arithmetic circuit satisfiability language of Bulletproofs, thus obtaining a construction under standard assumptions and without preprocessing. To achieve this, we slightly modify the arithmetic circuit proof protocol to cast its algebraic verifiers into the form required by ProtoStar. We then design the *Bulletproofs∗* (BP∗, BulletproofsStar) folding scheme that is complete and knowledge-sound. Finally, by applying the folding-to-IVC transformation of [12, 11], we estimate the verifier cost of the resulting IVC. This yields an asymptotic linear gain compared to repeated invocations of the monolithic Bulletproofs verifier.

1.2 Organization

Section 2 provides notation, security definitions, and cryptographic primitives. Section 3 is the related work on recursive proofs and folding schemes. Section 4 presents our folding scheme design, introduces the modified Bulletproofs arithmetic circuit proof, and derives the error terms required for folding; Section 5 presents the *Bulletproofs∗* folding scheme and analyzes its complexity; Section 6 evaluates *Bulletproofs∗* against the reference proof system; Section 7 gives the conclusions and future directions.

2 Preliminaries

2.1 Notations

With PPT we specify probabilistic polynomial-time algorithms and with λ the security parameter. We write $x \xleftarrow{\$} \mathcal{X}$ when we sample x uniformly at random from the set $\mathcal{X}$. We denote by $\mathbb{G}$ a cyclic group of prime order p. We consider G a generator of $\mathbb{G}$, $H \xleftarrow{\$} \mathbb{G}$ a distinct generator from G. We denote by $\mathbb{Z}_p$ the ring of integers modulo p and by $\mathbb{Z}_p^*$ the set of its invertible elements. We use additive notation for group operations in $\mathbb{G}$: the addition of two elements $A, B \in \mathbb{G}$ is $A + B$, the scalar multiplication is defined as $k \cdot A$ for any $k \in \mathbb{Z}_p$. We write $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_p^n$ for n-length vectors of scalars and $\mathbf{G} = (G_1, \ldots, G_n) \in \mathbb{G}^n$ for n-length vectors of generators. The inner product between two n-length scalar vectors $\mathbf{a}$ and $\mathbf{b}$ is $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i \cdot b_i \in \mathbb{Z}_p$, the Hadamard product between the same vectors is $\mathbf{a} \circ \mathbf{b} = (a_1 \cdot b_1, ..., a_n \cdot b_n) \in \mathbb{Z}_p^n$. Let $\mathbf{a} \in \mathbb{Z}_p^n$ and $\mathbf{G} \in \mathbb{G}^n$, the multi-scalar multiplication is defined as $\mathbf{a} \cdot \mathbf{G} = \langle \mathbf{a}, \mathbf{G} \rangle \in \mathbb{G}$. We write $\mathbf{A} \in \mathbb{Z}_p^{m \times n}$ for matrices with m rows and n columns and $a_{i,j}$ for elements of $\mathbf{A}$ in the i-th row and j-th column. Let $\mathbf{A} \in \mathbb{Z}_p^{m \times n}$, $\mathbf{a} \in \mathbb{Z}_p^n$ and $\mathbf{b} \in \mathbb{Z}_p^m$, we define $\mathbf{A} \cdot \mathbf{a} = (\sum_{j=1}^{n} w_{1,j} a_j, \ldots, \sum_{j=1}^{n} w_{m,j} a_j) \in \mathbb{Z}_p^m$ and $\mathbf{b} \cdot \mathbf{A} = (\sum_{i=1}^{m} b_i w_{i,1}, \ldots, \sum_{i=1}^{m} b_i w_{i,n}) \in \mathbb{Z}_p^n$. We denote by $\mathbf{a}_{[:k]} = (a_1, \ldots, a_k) \in \mathbb{Z}_p^k$ and $\mathbf{a}_{[k:]} = (a_{k+1}, \ldots, a_n) \in \mathbb{Z}_p^{n-k}$ the slices of $\mathbf{a} \in \mathbb{Z}_p^n$.

2.2 Special-sound arguments of knowledge

We revise the following definitions from [2, 26, 11, 22].

Definition 1 (Public-coin interactive arguments [11]). *Let $\mathcal{R}$ be a relation for statement x and witness w, defining an NP language $\mathcal{L}$. In particular, $x \in \mathcal{L}$ implies that there exists an efficiently verifiable witness w such that $(x, w) \in \mathcal{R}$. An interactive argument for $\mathcal{R}$ is a two-party conversation between two PPT algorithms, a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, with $\mathcal{P}$ and $\mathcal{V}$ taking as input the public statement x and only $\mathcal{P}$ the private witness w. The conversation produces a transcript $tr \leftarrow \langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle$. We say that the transcript is accepting if $\mathcal{V}$ outputs 1 after interacting with $\mathcal{P}$, and we denote this event by $\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle = 1$, rejecting otherwise. The acceptance is interpreted as $\mathcal{V}$ being convinced that $x \in \mathcal{L}$. In*

an interactive argument, completeness ensures that $\mathcal{V}$ accepts true statements, whereas soundness ensures that the verifier rejects false statements from any PPT prover $\mathcal{P}^*$ except with negligible probability. We say that an interactive argument is public-coin if the messages of $\mathcal{V}$ are only random values generated independently of the messages of $\mathcal{P}$. An interactive argument is $(2k+1)$-move if there are k verifier messages and $k+1$ prover messages.

Definition 2 (Interactive arguments of knowledge). *A public-coin interactive argument is an argument of knowledge for a relation $\mathcal{R}$ if it has perfect completeness and special-soundness.*

Definition 3 (Perfect completeness [26]). *An interactive argument for a relation $\mathcal{R}$ is perfectly complete if the verifier always accepts a conversation with a prover acting honestly.*

Definition 4 (t-Special-Soundness [2]). *A 3-move public-coin interactive argument for a relation $\mathcal{R}$ is t-special-sound if there exists a polynomial-time extractor $\mathcal{E}$ such that, given the statement x and t accepting transcripts (a, c_1, z_1), ..., (a, c_t, z_t) with the same prover's first message a and pairwise-distinct verifier's challenges $c_1, \ldots, c_t$, the extractor outputs a witness w such that $(x, w) \in \mathcal{R}$. When $t = 2$ we simply say that the argument is special-sound.*

A generalization of t-special-soundness for $(2k+1)$-move public-coin interactive arguments $(\mathcal{P}, \mathcal{V})$ is known as $(t_1, ..., t_k)$-*special-soundness* [7]. In this version, the polynomial-time extractor $\mathcal{E}$ takes as input the statement x and a $(t_1, ..., t_k)$-tree of accepting transcripts as in Definition 5, and outputs a witness satisfying $(x, w) \in \mathcal{R}$. Moreover, if such an efficient extraction procedure exists from a tree of transcripts with this structure, then $(\mathcal{P}, \mathcal{V})$ has *witness-extended emulation* by the forking lemma [20].

Definition 5 (Tree of transcripts [2]). *A $(t_1, ..., t_k)$-tree of transcripts for a $(2k+1)$-move public-coin interactive argument is a set of $T = \prod_{i=1}^{k} t_i$ transcripts organized in a tree structure. In this structure, each node corresponds to a prover's message (which may be empty) and each edge is labeled with a verifier's challenge. A node at depth i has precisely t_i children, each linked to a pairwise distinct challenge. The full tree contains T unique paths from root to leaf, with each path corresponding to an accepting transcript.*

2.3 Zero-knowledge arguments and Fiat-Shamir transform

An additional property of arguments of knowledge is the *zero-knowledge*. In this setting, the prover $\mathcal{P}$ aims to convince the verifier $\mathcal{V}$ of a statement's validity without revealing any additional information, particularly about the secret witness. The definition of zero-knowledge follows the simulation paradigm. We use the specific notion of *Special Honest-Verifier Zero-Knowledge* (SHVZK), that requires a simulator for the honest verifier and with the verifier's internal randomness known to the simulator in advance [10].

Definition 6 (Perfect Special Honest-Verifier Zero-Knowledge [26]). *A public-coin argument of knowledge $(\mathcal{P}, \mathcal{V})$ has perfect honest-verifier zero-knowledge if there exists a PPT algorithm $\mathcal{S}$, known as simulator, which, given as input the public statement x and the public-coin randomness of $\mathcal{V}$, produces transcripts whose distribution is the same as that of transcripts generated during the protocol executions of $(\mathcal{P}, \mathcal{V})$.*

We can convert a public-coin argument of knowledge into a Non-Interactive Argument of Knowledge (NARK) using the Fiat-Shamir transform [17]. This transformation replaces each random challenge with the hash of the transcript up to that point, thus yielding a NARK in the random oracle model [4]. The prover and verifier of a random-oracle NARK, i.e., $(\mathcal{P}^{\mathcal{O}_{NARK}}, \mathcal{V}^{\mathcal{O}_{NARK}})$, are called *$T$-query random-oracle algorithms* when they are given black-box access to a random oracle $\mathcal{O}_{NARK}$ and make at most T queries to it. The knowledge-soundness property for random-oracle NARKs necessarily depends on the query complexity T.

Definition 7 (Knowledge-Soundness [3]). *A random-oracle NARK $(\mathcal{P}^{\mathcal{O}_{NARK}}, \mathcal{V}^{\mathcal{O}_{NARK}})$ is knowledge-sound with knowledge error $\mu : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$, if there exists a knowledge extractor $\mathcal{E}$ with the following properties: $\mathcal{E}^{\mathcal{P}^*}(x)$, with input statement x and oracle access to a (potentially dishonest) T-query random-oracle prover $\mathcal{P}^*$, runs in an expected polynomial time in $|x|$ and T and outputs a witness w such that $(x, w) \in \mathcal{R}$ with probability*

$$Pr[(x, w) \in \mathcal{R} : w \leftarrow \mathcal{E}^{\mathcal{P}^*}(x)] \geq \frac{\epsilon(\mathcal{P}^*, x) - \mu(|x|, T)}{poly(|x|)},$$

where $\epsilon(\mathcal{P}^*, x) = Pr[\mathcal{V}^{\mathcal{O}_{NARK}}(x, \pi) = 1 : \pi \leftarrow \mathcal{P}^*]$. Here, $\mathcal{E}$ implements $\mathcal{O}_{NARK}$ for $\mathcal{P}^*$, in particular, $\mathcal{E}$ can arbitrarily program $\mathcal{O}_{NARK}$.

Adaptive knowledge-soundness for random-oracle NARKs differs from the non-adaptive knowledge-soundness in Definition 7 in that $\mathcal{P}^*$ outputs not only a proof π but also a statement x, i.e., $\mathcal{P}^*$ is free to choose x adaptively. In this setting, the extractor is required to output $(x, \pi, v; w)$, where $v \in \{0, 1\}$ indicates whether π is a valid proof for x and (x, π, v) is identically distributed to $(x, \pi, v) \leftarrow \mathcal{P}^*$. Then, the success probability of the extractor depends on the success probability $\epsilon(\mathcal{P}^*)$ of the adaptive prover $\mathcal{P}^*$. To support compositional settings, $\mathcal{P}^*$ may also output auxiliary information aux, while the extractor is required to simulate the tuple (x, π, aux, v) [3].

The Fiat-Shamir transformation is adaptive when the statement x is also hashed with the transcript (e.g., it may be included in the first challenge). A positive result in [3] establishes an upper bound on the knowledge error of Fiat-Shamir transformed $(2k + 1)$-move public-coin interactive arguments.

Lemma 1 (Fiat-Shamir transform of Special-Sound Arguments [11]). *The Fiat-Shamir transform of a $(t_1, ..., t_k)$-out-of-N, with N-sized challenge*

space, special-sound interactive argument is knowledge sound with knowledge error

$$\mu_{fs}(T) = (T+1)\mu$$

where $\mu = 1 - \prod(1 - \frac{t_i}{N})$ *is the knowledge error of the interactive argument.*

2.4 Assumptions and commitments

We work in prime-order cyclic groups where the discrete logarithm problem is assumed to be computationally intractable (DLOG assumption).

Definition 8 (Non-trivial DLOG relation [10]). *For any PPT adversary* $\mathcal{A}$ *and any* $j \geq 2$*, there exists a negligible function* $\text{negl}(\lambda)$*, such that*

$$Pr\left[\begin{array}{l} G_1, ..., G_j \xleftarrow{\$} \mathbb{G}; \quad b_1, ..., b_j \in \mathbb{Z}_p \leftarrow \mathcal{A}(\mathbb{G}, G_1, ..., G_j) \\ : \quad b_1, ..., b_j \neq 0 \ \wedge \ \sum_{i=1}^{j} b_i G_i = 0 \end{array}\right] \leq \text{negl}(\lambda)$$

In other words, the probability that $\mathcal{A}$ finds a non-trivial DLOG relation between uniformly random group elements is negligible.

We use *Pedersen commitments* [26] with a commitment space $\mathbb{G}$, and a message space $\mathbb{Z}_p$. Let $G, H \in \mathbb{G}$, $\mathbf{G} \in \mathbb{G}^n$, $m \in \mathbb{Z}_p$ and $r \xleftarrow{\$} \mathbb{Z}_p^*$ (a random blinding term). A commitment C to m is computed as

$$C = mG + rH \in \mathbb{G}.$$

If $\mathbf{m} \in \mathbb{Z}_p^n$, a vector commitment C to $\mathbf{m}$ is computed as

$$C = \langle \mathbf{m}, \mathbf{G} \rangle + rH \in \mathbb{G}.$$

Let $C_1, C_2 \in \mathbb{G}$ be two commitments to $m_1, m_2 \in \mathbb{Z}_p$ respectively, and $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p^*$. Pedersen commitments are *homomorphic*, thus we can compute

$$C_1 + C_2 = (m_1 + m_2)G + (r_1 + r_2)H \in \mathbb{G}.$$

Pedersen commitments are *binding* and *hiding* under the DLOG assumption. Binding means that it is computationally hard to find two openings for a commitment. Hiding means that it is hard to distinguish which is the committed message. If no blinding term is used, the resulting commitment is binding but not hiding.

We sometimes write a commitment function

$$Commit(\mathbf{m}_1, \mathbf{m}_2, r) := \langle \mathbf{m}_1, \mathbf{G} \rangle + \langle \mathbf{m}_2, \mathbf{H} \rangle + rH,$$

that maps to the relative generator (or vector of generators) according to the input.

2.5 Folding schemes

We recall the notion of folding schemes from [11], which ProtoStar also refers to as accumulation schemes. In particular, what gets folded are two NARK proofs: one associated with the running instance and one carried by an accumulator. Accordingly, a folding scheme for a NARK $(\mathcal{P}_{NARK}, \mathcal{V}_{NARK})$ is a triple of algorithms, prover, verifier, and decider $(\mathcal{P}_{\text{fold}}, \mathcal{V}_{\text{fold}}, \mathcal{D})$, all with access to the same random oracles $\mathcal{O}_{NARK}$,$\mathcal{O}_{fold}$:

- $\mathcal{P}_{\text{fold}}(\text{pi}, \pi = (\pi_x, \pi_w), \text{acc} = (\text{acc}_x, \text{acc}_w)) \to \{\text{acc}' = (\text{acc}'_x, \text{acc}'_w), \text{pf}\}$, where pi is the statement, π the NARK proof, acc the accumulator, acc$'$ the new accumulator, pf the accumulation proof.
- $\mathcal{V}_{\text{fold}}(\text{pi}, \pi_x, \text{acc}_x, \text{acc}'_x, \text{pf}) \to \{0, 1\}$, where pi is the statement, π_x the NARK proof instance, acc_x and acc'_x the old and new accumulator instances, pf the accumulation proof. The verifier outputs 1 (accept) or 0 (reject).
- $\mathcal{D}(\text{acc}) \to \{0, 1\}$. The decider on input the accumulator outputs 1 (accept) or 0 (reject).

A folding scheme is perfectly complete and knowledge-sound with knowledge error μ if the random-oracle NARK $(\mathcal{P}', \mathcal{V}')$ is perfectly complete and knowledge-sound with knowledge error μ for the relation

$$\mathcal{R}_{fold} = \{(\text{pi}, \pi_x, \text{acc}_x;\ \pi_w, \text{acc}_w) : \mathcal{V}_{NARK}(\text{pi}, \pi) = 1\ \wedge\ \mathcal{D}(acc) = 1\}. \tag{1}$$

where $\mathcal{P}'$ outputs the tuple (acc$'$, pf) and $\mathcal{V}'$, on input ((pi, π_x, acc_x), (acc$'$, pf)), accepts if $\mathcal{D}(\text{acc}') = 1$ and $\mathcal{V}_{\text{fold}}(\text{pi}, \pi_x, \text{acc}_x, \text{acc}'_x, \text{pf}) = 1$.

2.6 Bulletproofs arithmetic circuits

We recall the arithmetic circuit relation of Bulletproofs [10]:

$$\begin{aligned}
&\mathcal{R}_{AC} = \{(G, H \in \mathbb{G}, \mathbf{G}, \mathbf{H} \in \mathbb{G}^n, \mathbf{V} \in \mathbb{G}^m, \mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O \in \mathbb{Z}_p^{Q \times n},\\
&\mathbf{W}_V \in \mathbb{Z}_p^{Q \times m}, \mathbf{c} \in \mathbb{Z}_p^Q;\ \mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O \in \mathbb{Z}_p^n, \mathbf{v}, \boldsymbol{\gamma} \in \mathbb{Z}_p^m) :\\
&V_j = v_j G + \gamma_j H\ \forall j \in [1, m]\ \wedge\ \mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O\ \wedge\\
&\mathbf{W}_L \cdot \mathbf{a}_L + \mathbf{W}_R \cdot \mathbf{a}_R + \mathbf{W}_O \cdot \mathbf{a}_O = \mathbf{W}_V \cdot \mathbf{v} + \mathbf{c}\}.
\end{aligned} \tag{2}$$

Hence, an arithmetic circuit with n multiplication gates can be converted into a Hadamard-product relation with additional $Q \leq 2n$ linear constraints that include the opening of commitments V_j.

3 Related Work

Eli Ben-Sasson et al. [5] implement the first scalable preprocessing (trusted setup) zk-SNARK that is practical for recursive proof composition by exploiting cycles of elliptic curves. Such result highlights how suitably combined elliptic

curves lie on a PCD-friendly[3] cycle; a known example is the PCD-friendly 2-cycle of two elliptic curves where the base field size of one curve equals the group order of the other curve (and viceversa). In this setting, it is possible to construct two zk-SNARKs on the two curves, and use them to enable efficient recursion. In particular, each zk-SNARK prove statements about the verifier of the other zk-SNARK. By alternating the two proof systems, each verification step is expressed in the field that the other system can handle natively.

Halo [8] enables recursive proof composition without a trusted setup under standard assumptions and over cycles of elliptic curves that are not pairing-friendly. First, Halo presents a Polynomial Commitment Scheme (PCS) based on a modified Inner Product Argument (IPA) relation of [7, 10] that is tied to a variant of Sonic [21] proof system to work with the PCS. The resulting argument for arithmetic circuit satisfiability exploits polynomial commitment amortization techniques, and a nested amortization strategy is used to enable recursive proof composition. The verification circuit avoids performing any linear-time work, instead, it treats the deferred check as part of the statement by taking as public inputs both the input and the prover's claimed output of the linear-time computation. This allows the verifier to merge two instances of the statement into a single instance, thus amortizing the cost of the expensive operation across many composed proofs. Finally, only at the end of the recursion, the ultimate verifier executes the linear-time check to validate the entire sequence of proofs.

Proof-Carrying Data (PCD) [13] builds on the previous result and shows that there is a general paradigm to construct *accumulation schemes* for succinct non-interactive arguments of knowledge (SNARKs) even with non-sublinear-time verifier. In particular, an accumulation scheme can be derived for any SNARK whose verifier is succinct except for the evaluation of a specific statement. Moreover, if the specific statements are opening of polynomial commitments, then two PCS, one based on DLOG assumptions and the other based on knowledge assumptions in bilinear groups, have accumulation schemes in the random oracle model. Finally, the authors prove that any SNARK with a sublinear accumulation verifier yields a PCD scheme.

Halo Infinite [6] further generalizes the previous result, showing that PCD can be built heuristically from any homomorphic PCS, even when evaluation proofs are not sublinear in either proof size (w.r.t. the polynomial degree) or verification time. In particular, the authors observe that Halo's approach applies to any PCS that supports linear combinations of commitments, which defines a Linear Combination Scheme (LCS). Moreover, the authors outline the potential of a PCS to aggregate evaluation proofs to amortize the verifier work. This lands to two generalized results: i) every PCS with efficient LCS has an efficient private aggregation, i.e., an interactive proof that combines many evaluation proof instances into one such that accepting the aggregate implies validity of all instances; ii) every succinct additive PCS has an efficient public aggregation, i.e., many evaluation proofs can be aggregated non-interactively into one instance,

[3] Since they facilitate the construction of PCD, i.e., a primitive that encapsulates the recursive proof composition.

even without knowing witnesses for the input instances, and a valid proof for the output instance implies validity of each of the input proofs. Overall, aggregation enables batch evaluation of PCS and yields generalized recursive proofs/PCD constructions: PCD with an efficient prover can be obtained from PCS admitting an efficient LCS, PCD with sublinear proofs follows from PCS with an efficient public aggregation.

PCD "without succinct arguments" [12] shows that a PCD can be constructed from any non-interactive argument of knowledge (NARK) that satisfies the notion of *split accumulation*. A split accumulator divides a classical accumulator into an instance part and a witness part, and the accumulation verifier only receives the instance part. This enables to design a succinct accumulation verifier with short instances and not large witnesses. Building on this notion, the authors show that any NARK having a split accumulation with a sublinear accumulation verifier yields a PCD, even if NARK does not have sublinear argument size. Focusing on minimizing the cost of the accumulation verifier, the authors develop split accumulation for a transparent NARK for R1CS and for predicates such as Hadamard products and polynomial evaluations based on Pedersen commitments.

Nova [19] introduces a (trustless and DLOG-based) primitive called *folding scheme*, which can be used to reduce the task of checking two instances in some relation to the task of checking a single (folded) instance. Unlike prior SNARK-based IVC approaches, which accumulate expensive steps of the verifier circuit, Nova accumulates the instances themselves. Concretely, Nova instantiates a folding scheme for arithmetic circuit satisfiability by introducing *relaxed R1CS* and relying on succinct homomorphic commitments (e.g., Pedersen commitments). Nova then builds an IVC from a non-interactive folding scheme for relaxed R1CS: all steps of the incremental computation are expressed with R1CS (sharing the same coefficient matrices), and at each step Nova prover proves that it has executed the step computation and has folded its prior step into a running relaxed R1CS instance. In the incremental steps, the (recursively composed) verifier circuit verifies the prover correctly folds the previous step. The resulting IVC proof size and verifier time are linear in the size of the computation, but they can be optimized using zkSNARKs.

4 Bulletproofs∗ design

In this section, we give an overview of ProtoStar, describe how we adapt the algebraic verifier of Bulletproofs to the ProtoStar form, and show how our prover computes the error terms required by the folding scheme design.

4.1 ProtoStar and negative exponents

ProtoStar works for public-coin $(2k+1)$-move interactive arguments $(\mathcal{P}, \mathcal{V})$ whose verifier, after k challenges, computes an algebraic map $\mathcal{V}$ of degree D and checks that the output is a zero vector of length l. For a transcript $((\mathbf{m_i}, r_i)_{i=1}^{k}, \mathbf{m}_{k+1})$

of prover messages $(\mathbf{m_i})_{i=1}^{k+1}$ and verifier challenges $(r_i)_{i=1}^{k}$, the map $\mathcal{V}$ is defined as the sum of $D+1$ homogeneous degree-d algebraic maps:

$$\mathcal{V}(\text{pi}, \mathbf{m}_1, \ldots, \mathbf{m}_{k+1}, \mathbf{r}) := \sum_{d=0}^{D} f_d^{\mathcal{V}}(\text{pi}, \mathbf{m}_1, \ldots, \mathbf{m}_{k+1}, \mathbf{r}) = \mathbf{0}. \tag{3}$$

Let $(\mathcal{P}_{NARK}, \mathcal{V}_{NARK})$ be the NARK of the $(2k+1)$-move interactive argument $(\mathcal{P}, \mathcal{V})$, where the verifier challenges are derived via a random oracle, and π the NARK proof. The "relaxed" verifier checks introduce a slack variable σ and a vector of error terms $\mathbf{e}$ of length l:

$$\mathcal{V}(\text{pi}, \pi.\mathbf{m}_1, \ldots, \pi.\mathbf{m}_{k+1}, \mathbf{r}, \sigma) := \sum_{d=0}^{D} \sigma^{D-d} \cdot f_d^{\mathcal{V}}(\text{pi}, \pi.\mathbf{m}_1, \ldots, \pi.\mathbf{m}_{k+1}, \mathbf{r}) = \mathbf{e}. \tag{4}$$

Then, in the folding scheme the prover accumulates "relaxed" verifier checks of the NARK by computing:

$$\sum_{d=0}^{D} (X+\sigma)^{D-d} \cdot f_d^{\mathcal{V}}(X \cdot \text{pi} + \text{pi}', (X \cdot \mathbf{m}_i + \mathbf{m}_i')_{i=1}^{k+1}, (X \cdot r_i + r_i')_{i=1}^{k}), \tag{5}$$

which interpolates statement and NARK proof with the old accumulator and brings up the error terms that are used as an accumulation proof. Indeed, Equation (5) turns out to be, with $\mathcal{V}_{NARK} = 0$, equal to:

$$\mathbf{e} + \sum_{j=1}^{D-1} \mathbf{e}_j X^j + \mathcal{V}_{NARK}(\text{pi}, (\mathbf{m}_i)_{i=1}^{k+1}, (r_i)_{i=1}^{k}) X^D. \tag{6}$$

The calculation of error terms is a critical step in the design of a folding scheme, since the verifier's checks must have a special form, i.e., $\mathcal{V}$ is the sum of $D+1$ homogeneous degree-d algebraic maps. In any public-coin $(2k+1)$-move interactive argument with an algebraic verifier, when $\mathcal{V}$ checks equations with negative exponents, then the accumulated instances cannot be computed, as illustrated by the following example. Consider the map $f(r) = r^{-1}$, this results $f(Xr + r') = (Xr + r')^{-1}$, which does not imply an algebraic $\mathcal{V}$ of the right form $a_0 + a_1 X + \cdots + a_d X^d$ for any finite degree d and coefficients $(a_i)_{i=0}^{d}$.

4.2 Modified arithmetic circuit proof and homogeneous maps

We introduce a modified arithmetic circuit proof of Bulletproofs satisfying Relation (2). At a high level, the arithmetic circuit proof of Bulletproofs reduces the Hadamard-product relation, together with the linear constraints, to a single inner-product relation. Our protocol follows from the argument of Bulletproofs, except that we replace the verifier check enforcing the constraints with an equivalent check that avoids negative exponents. The full modified interactive argument $(\mathcal{P}, \mathcal{V})$ is shown in Figure 1.

$(\mathcal{P}, \mathcal{V})$'s **input**: $(G, H, \mathbf{G}, \mathbf{H}, \mathbf{V}, \mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O, \mathbf{W}_V, \mathbf{c};\ \mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O, \mathbf{v}, \boldsymbol{\gamma})$

$\mathcal{V}$'s **output**: $\{0, 1\}$

$\mathcal{P}$:

  $\gamma_I, \gamma_O, \gamma_S \xleftarrow{\$} \mathbb{Z}_p$

  $\mathbf{s}_L, \mathbf{s}_R \xleftarrow{\$} \mathbb{Z}_p^n$

  $A_I = \gamma_I H + \langle \mathbf{a}_L, \mathbf{G} \rangle + \langle \mathbf{a}_R, \mathbf{H} \rangle$

  $A_O = \gamma_O H + \langle \mathbf{a}_O, \mathbf{G} \rangle$

  $S = \gamma_S H + \langle \mathbf{s}_L, \mathbf{G} \rangle + \langle \mathbf{s}_R, \mathbf{H} \rangle$

$\mathcal{P} \to \mathcal{V} : A_I, A_O, S$

$\mathcal{V} \to \mathcal{P} : y, z \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P}$ and $\mathcal{V}$:

  $\mathbf{z}_{[1:]}^{Q+1} = (z, z^2, \ldots, z^Q)$

  $\mathbf{y}^n = (1, y, y^2, \ldots, y^{n-1})$

  $\delta_{new}(y, z) = \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L \rangle$

$\mathcal{P}$:

  $l(X) = X(\mathbf{a}_L + \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R)) + X^2 \mathbf{a}_O + X^3 \mathbf{s}_L$

  $r(X) = -\mathbf{y}^{-n} + \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O + X(\mathbf{y}^{-n} \circ \mathbf{a}_R + \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L) + X^3(\mathbf{y}^{-n} \circ \mathbf{s}_R)$

  $t(X) = \langle l(X), r(X) \rangle$

  $w = \langle \mathbf{a}_L, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L \rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R, \mathbf{a}_R \rangle + \langle \mathbf{a}_O, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O \rangle$

  $t_2 = \langle \mathbf{a}_L, \mathbf{y}^{-n} \circ \mathbf{a}_R \rangle - \langle \mathbf{a}_O, \mathbf{y}^{-n} \rangle + w + \delta_{new}(y, z)$

  $\rho_{1,3,4,5,6} \xleftarrow{\$} \mathbb{Z}_p$

  $T_{1,3,4,5,6} \leftarrow t_{1,3,4,5,6} G + \rho_{1,3,4,5,6} H$

$\mathcal{P} \to \mathcal{V} : T_1, T_3, T_4, T_5, T_6$

$\mathcal{V} \to \mathcal{P} : x \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P}$:

  $\tau_x = \rho_1 x + \sum_{i=3}^{6} \rho_i x^i + x^2 \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{W}_V \cdot \boldsymbol{\gamma} \rangle$

  $\mu = \gamma_I x + \gamma_O x^2 + \gamma_S x^3$

  $\hat{t} = t(x)$

  $\mathbf{l} = l(x)$

  $\mathbf{r} = r(x)$

$\mathcal{P} \to \mathcal{V} : \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}$

$\mathcal{V}$:

  $\hat{t} \stackrel{?}{=} \langle \mathbf{l}, \mathbf{r} \rangle$ check the correctness of $\hat{t}$

  $\hat{t}G + \tau_x H \stackrel{?}{=} x^2 \delta_{\text{new}}(y, z) G + x^2 \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle G + x^2 \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle$

  $+ x T_1 + \sum_{i=3}^{6} x^i T_i$ check the constraints

  $x A_I + x^2 A_O + x^3 S - \langle \mathbf{1}^n, \mathbf{H} \rangle + x \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L), \mathbf{H} \rangle$

37: $+x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{G}\rangle + \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O), \mathbf{H}\rangle \stackrel{?}{=}$

38: $\mu H + \langle \mathbf{l}, \mathbf{G}\rangle + \langle \mathbf{y}^n \circ \mathbf{r}, \mathbf{H}\rangle$ check the consistency of commitments

Figure 1: Modified arithmetic circuit interactive argument $(\mathcal{P}, \mathcal{V})$.

Specifically, we run into the aforementioned problem when we attempt to accumulate the verifier check enforcing the constraints, since the term $\delta(y, z)$ in the original argument maps the verifier challenge y to $\mathbf{y}^{-n}$. The same issue arises when we attempt to accumulate the verifier check enforcing the consistency of commitments, due to the generators $\mathbf{H}'$ and the weights W_L, W_R, W_O. For the complete verifier checks in the original argument, we refer the reader to checks (89) and (92) in [10]. We solve this problem by introducing two new algebraic verifiers $\mathcal{V}^{\delta_{\text{new}}}$ and $\mathcal{V}^{comm}$ without negative exponents, obtained by substituting $\mathbf{y}^n \to l(x)$ and $\mathbf{y}^{-n} \to r(x)$. In particular, we have that:

$$\langle l(x), r(x)\rangle = \langle x(\mathbf{a}_L + \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R)) + x^2\mathbf{a}_O + x^3\mathbf{s}_L, -\mathbf{y}^{-n} + \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O$$
$$+x(\mathbf{y}^{-n} \circ \mathbf{a}_R + \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L) + x^3(\mathbf{y}^{-n} \circ \mathbf{s}_R)\rangle.$$

From this inner product, we derive the term

$$t_2 = \langle \mathbf{a}_L, \mathbf{y}^{-n} \circ \mathbf{a}_R\rangle - \langle \mathbf{a}_O, \mathbf{y}^{-n}\rangle$$
$$+\langle \mathbf{a}_L, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L\rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R, \mathbf{a}_R\rangle + \langle \mathbf{a}_O, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O\rangle$$
$$+\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L\rangle,$$

where $\delta_{new}(y, z) = \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L\rangle$ does not contain negative exponents.

Hence, we define the algebraic verifier $\mathcal{V}^{\delta_{\text{new}}}$

$$\begin{aligned}
&\mathcal{V}^{\delta_{\text{new}}}(\mathbf{V}, \mathbf{c}, T_1, T_3, \ldots, T_6, \tau_x, \hat{t}, y, z, x) := \\
&\hat{t}G + \tau_x H - x^2\delta_{\text{new}}(y, z)G - x^2\langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c}\rangle G - x^2\langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V}\rangle \\
&- xT_1 - \sum_{i=3}^{6} x^i T_i = 0,
\end{aligned} \tag{7}$$

while the algebraic verifier $\mathcal{V}^{comm}$ follows from the new vectors $\mathbf{l}$ and $\mathbf{r}$ and the updated basis $\mathbf{H}' = \mathbf{y}^n \circ \mathbf{H}$. Expressed entirely in basis $\mathbf{H}$, $\mathcal{V}^{comm}$ becomes

$$\begin{aligned}
&\mathcal{V}^{comm}(A_I, A_O, S, \mu, \mathbf{l}, \mathbf{r}, y, z, x) = \\
&xA_I + x^2A_O + x^3S - \langle \mathbf{1}^n, \mathbf{H}\rangle + x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L), \mathbf{H}\rangle \\
&+ x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{G}\rangle + \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O), \mathbf{H}\rangle \\
&- \mu H - \langle \mathbf{l}, \mathbf{G}\rangle - \langle \mathbf{y}^n \circ \mathbf{r}, \mathbf{H}\rangle = 0.
\end{aligned} \tag{8}$$

Both Equation (7) and Equation (8) are in the form of Equation (3), where $\mathbf{V} = (V_j)_{j=1}^{m}$ and $\mathbf{c} = (c_i)_{i=1}^{Q}$ are in pi, $(A_I, A_O, S,\ T_1, T_3, \ldots, T_6,\ \tau_x, \mu, \hat{t},\ \mathbf{l}, \mathbf{r})$ are in $(\mathbf{m}_i)_{i=1}^{k+1}$ and (y, z, x) in $(r_i)_{i=1}^{k}$. Note that $G, H,\ \mathbf{G}, \mathbf{H},\ \mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O$ and $\mathbf{W}_V$ from Relation (2) are not included in pi, since the CRS and circuit description parameters are not part of the accumulated instances.

The following theorem shows that the modified arithmetic circuit proof of Bulletproofs, presented in Figure 1, preserves security.

Theorem 1 (Modified arithmetic circuit proof (Bulletproofs)). *The modified arithmetic circuit proof of Bulleproofs has perfect completeness, perfect honest-verifier zero-knowledge and special-soundness satisfying Relation (2).*

Proof sketch. The proof proceeds analogously to the Bulletproofs proof. Completeness holds since t_2 is derived correctly and an honest prover conversation is accepted. Zero-knowledge follows by the existence of a simulator that, given the statement and the public-coin randomness, outputs a transcript $(A_I, A_O, S, y, z, T_1, T_3, \ldots, T_6, x, \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r})$ that is identically distributed to the transcript produced in an honest execution between prover and verifier. To prove special-soundness, we can use the same extractor of Bulletproofs, which rewinds the prover and processes a poly-bounded (in λ) set of accepting transcripts to extract either a valid witness $(\mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O, \mathbf{v}, \boldsymbol{\gamma})$ or a non-trivial DLOG relation. This concludes the proof sketch.

The complete proof appears in Appendix A.

Following the ProtoStar's recipe, Equation (7) can be expressed as a sum of homogeneous degree-d algebraic maps $f_d^{\mathcal{V}^{\delta_{\text{new}}}}(\mathbf{V}, \mathbf{c}, T_1, T_3, \ldots, T_6, \tau_x, \hat{t}, y, z, x)$, for $0 \leq d \leq n + 1 + 2Q$. The upper bound follows form the maximum polynomial degree contributed by the term $-x^2\delta_{\text{new}}(y, z)G$. In what follows, we give the full decomposition of $\mathcal{V}^{\delta_{\text{new}}}$ into homogeneous maps with implicit parameters.

$$
\begin{aligned}
f_0^{\mathcal{V}^{\delta_{\text{new}}}} &= 0,\\
f_1^{\mathcal{V}^{\delta_{\text{new}}}} &= \hat{t}G + \tau_x H,\\
f_2^{\mathcal{V}^{\delta_{\text{new}}}} &= -xT_1,\\
f_3^{\mathcal{V}^{\delta_{\text{new}}}} &= 0,\\
f_{4\leq d\leq 7}^{\mathcal{V}^{\delta_{\text{new}}}} &= -x^{d-1}T_{d-1} - x^2 z^{d-3} c_{d-3} G\\
&\quad -x^2 \sum_{k=1}^{m} z^{d-3} w_{V_{d-3,k}} V_k - x^2 \Big(\sum_{i=2}^{d-2} k_{i-1,d-i} y^{i-2} z^{d-i}\Big) G,
\end{aligned}
$$

where $k_{j,s} = \sum_{i=1}^{Q} w_{R_{i,j}} w_{L_{s-i,j}}$ with $w_{L_{s-i,j}} = 0$ if $s - i \notin [1, Q]$. Finally, the maps $f_d^{\mathcal{V}^{\delta_{\text{new}}}}$ for $7 < d \leq Q + 3$ follow from the previous but omit the contribution of the terms $-\sum_{i=3}^{6} x^i T_i$, while the maps $f_d^{\mathcal{V}^{\delta_{\text{new}}}}$ for $Q+3 < d \leq n+1+2Q$

consider only $-x^2\delta_{\text{new}}(y,z)G$.

In a similar way, we can express Equation (8) as a sum of homogeneous degree-d maps $f_d^{\mathcal{V}^{comm}}(A_I, A_O, S, \mu, \mathbf{l}, \mathbf{r}, y, z, x)$, for $0 \leq d \leq n+Q$. The upper bound is given by the highest degree coming from terms $+x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L), \mathbf{H}\rangle$ and $+x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{G}\rangle$. In what follows, we give the full decomposition of $\mathcal{V}^{comm}$ into homogeneous maps with implicit parameters. First, for $1 \leq d \leq n+Q$, we define the common map:

$$g_d := \sum_{j=1}^{n}\sum_{i=1}^{Q}(xy^{j-1}z^i w_{L_{i,j}})H_j + \sum_{j=1}^{n}\sum_{i=1}^{Q}(xy^{j-1}z^i w_{R_{i,j}})G_j$$

$$+\sum_{j=1}^{n}\sum_{i=1}^{Q}(y^{j-1}z^i w_{O_{i,j}})H_j - \sum_{j=1}^{n} y^{j-1}r_j H_j,$$

with $w_{L_{i,j}} = w_{R_{i,j}} = 0$ if $i+j \neq d$, $w_{O_{i,j}} = 0$ if $i+j-1 \neq d$ and $r_j = 0$ if $j \neq d$. Then, we have the full decomposition:

$$f_0^{\mathcal{V}^{comm}} = -\langle \mathbf{1}^n, \mathbf{H}\rangle,$$

$$f_1^{\mathcal{V}^{comm}} = -\mu H - \langle \mathbf{l}, \mathbf{G}\rangle + g_1,$$

$$f_2^{\mathcal{V}^{comm}} = xA_I + g_2,$$

$$f_3^{\mathcal{V}^{comm}} = x^2 A_O + g_3,$$

$$f_4^{\mathcal{V}^{comm}} = x^3 S + g_4,$$

$$f_{4<d\leq n}^{\mathcal{V}^{comm}} = g_d,$$

and the maps $f_d^{\mathcal{V}^{comm}}$ for $n < d \leq n+Q$ follow from the previous but omit the contribution of the terms $-\langle \mathbf{y}^n \circ \mathbf{r}, \mathbf{H}\rangle$.

It remains the verifier check that enforces the correctness of $\hat{t}$, which does not require any adjustment due to negative exponents. We write the algebraic verifier $\mathcal{V}^{\hat{t}}$ in the form of Equation (3):

$$\mathcal{V}^{\hat{t}}(\hat{t}, \mathbf{l}, \mathbf{r}) := \hat{t} - \langle \mathbf{l}, \mathbf{r}\rangle = 0 \tag{9}$$

Then, Equation (9) can be expressed as a sum of homogeneous degree-d algebraic maps $f_d^{\mathcal{V}^{\hat{t}}}(\hat{t}, \mathbf{l}, \mathbf{r})$, for $0 \leq d \leq 2$. The full decomposition of $\mathcal{V}^{\hat{t}}$ into homogeneous maps with implicit parameters is: $f_0^{\mathcal{V}^{\hat{t}}} = 0$, $f_1^{\mathcal{V}^{\hat{t}}} = \hat{t}$, $f_2^{\mathcal{V}^{\hat{t}}} = -\langle \mathbf{l}, \mathbf{r}\rangle$.

4.3 Computing the error terms

The prover accumulates "relaxed" verifier checks by computing the error terms via Equation (5). Let $\mathcal{V}$ be an algebraic verifier of degree D, in the form of a sum of homogeneous degree-d maps $f_d^{\mathcal{V}}$, and define interpolated inputs as $u''(X) := Xu + u'$. Here, u can be any element of the statement or of the NARK proof, and u' is the corresponding element of the accumulator. For a slack σ, we have that:

$$\sum_{d=0}^{D}(X+\sigma)^{D-d}\cdot f_d^{\mathcal{V}}(u''(X)) =$$

$$\sum_{d=0}^{D}\left(\sum_{a=0}^{D-d}\binom{D-d}{a}\sigma^{D-d-a}X^a\right)\cdot\left(\sum_{t=0}^{d}f_{d,t}^{\mathcal{V}}X^t\right) =$$

$$\sum_{j=0}^{D}\left(\sum_{d=0}^{D}\sum_{a=0}^{D-d}\binom{D-d}{a}\sigma^{D-d-a}f_{d,j-a}^{\mathcal{V}}\right)X^j,$$

where $f_{d,j-a}^{\mathcal{V}} = 0$ if $j-a \notin [0,d]$.

It turns out that, for $j = 1,\ldots,D-1$, the intermediate-degree coefficients are the error terms

$$\mathbf{e}_j = \sum_{d=0}^{D}\sum_{a=0}^{D-d}\binom{D-d}{a}\sigma^{D-d-a}f_{d,j-a}^{\mathcal{V}},$$

and the constant coefficient is $\mathbf{e} = \sum_{d=0}^{D}\sigma^{D-d}f_{d,0}^{\mathcal{V}}$.

In our design, the prover accumulates error terms for the algebraic verifiers $\mathcal{V}^{\delta_{\text{new}}}$, $\mathcal{V}^{comm}$ and $\mathcal{V}^{\hat{t}}$ of the NARK, i.e., Equation (7), Equation (8) and Equation (9) respectively. For each algebraic verifiers, the error terms are computed by deriving and collecting the coefficients $f_{d,t}^{\mathcal{V}^{\delta_{\text{new}}}}$, $f_{d,t}^{\mathcal{V}^{comm}}$ and $f_{d,t}^{\mathcal{V}^{\hat{t}}}$ as in the above procedure. In what follows, we provide formulas for deriving these coefficients, but the exact expressions may vary depending on the implementation.

Consider Equation (7) and let

$$\mathbf{V}''(X) = (V_j''(X))_{j=1}^m = (XV_j + V_j')_{j=1}^m,$$
$$\mathbf{c}''(X) = (c_i''(X))_{i=1}^Q = (Xc_i + c_i')_{i=1}^Q,$$
$$T_1''(X) = XT_1 + T_1',\ T_3''(X) = XT_3 + T_3', \ldots, T_6''(X) = XT_6 + T_6',$$
$$\tau_x''(X) = X\tau_x + \tau_x',\ \hat{t}''(X) = X\hat{t} + \hat{t}',$$
$$y''(X) = Xy + y',\ z''(X) = Xz + z',\ \text{and } x''(X) = Xx + x'.$$

By applying Equation (5) with these inputs, we derive the full list of coefficients $f_{d,t}^{\mathcal{V}^{\delta_{\text{new}}}}$, for $0 \le d \le n+1+2Q$ and $0 \le t \le d$:

$$f_{0,0}^{\mathcal{V}^{\delta_{\text{new}}}} = 0,$$

$$f_{1,0}^{\mathcal{V}^{\delta_{\text{new}}}} = \hat{t}'G + \tau_x' H, \quad f_{1,1}^{\mathcal{V}^{\delta_{\text{new}}}} = \hat{t}G + \tau_x H,$$

$$f_{2,0}^{\mathcal{V}^{\delta_{\text{new}}}} = -x'T_1', \quad f_{2,1}^{\mathcal{V}^{\delta_{\text{new}}}} = -(xT_1' + x'T_1), \quad f_{2,2}^{\mathcal{V}^{\delta_{\text{new}}}} = -xT_1,$$

$$f_{3,t}^{\mathcal{V}^{\delta_{\text{new}}}} = 0,$$

$$f_{(4\le d\le 7),t}^{\mathcal{V}^{\delta_{\text{new}}}} = -\binom{d-1}{t-1}x^{t-1}(x')^{d-t}T_{d-1} - \binom{d-1}{t}x^t(x')^{d-1-t}T_{d-1}'$$

$$- \sum_{\substack{u\in[0,2]\\ v\in[0,d-3]\\ w\in\{0,1\}\\ u+v+w=t}} \binom{2}{u}\binom{d-3}{v}x^u(x')^{2-u}z^v(z')^{d-3-v}(c_{d-3})^w(c_{d-3}')^{1-w}G$$

$$-\sum_{k=1}^{m} w_{V_{d-3,k}} \sum_{\substack{u\in[0,2]\\ v\in[0,d-3]\\ w\in\{0,1\}\\ u+v+w=t}} \binom{2}{u}\binom{d-3}{v}x^u(x')^{2-u}z^v(z')^{d-3-v}V_k^w(V_k')^{1-w}$$

$$-(\sum_{j=2}^{d-2} k_{j-1,d-j} \sum_{\substack{u\in[0,2]\\ v\in[0,j-2]\\ w\in[0,d-j]\\ u+v+w=t}} \binom{2}{u}\binom{j-2}{v}\binom{d-j}{w}x^u(x')^{2-u}y^v(y')^{j-2-v}z^w(z')^{d-j-w})G,$$

and $f_{d,t}^{\mathcal{V}^{\delta_{\text{new}}}}$ for $7 < d \le Q+3$ follow from the previous but without the folded terms of $-\sum_{i=3}^{6} x^i T_i$, while $f_{d,t}^{\mathcal{V}^{\delta_{\text{new}}}}$ for $Q+3 < d \le n+1+2Q$ consider only the folded terms of $-x^2\delta_{\text{new}}(y,z)G$.

Consider Equation (8) and let

$$A_I'' = XA_I + A_I', \ A_O'' = XA_O + A_O', \ S'' = XS + S',$$

$$\mu'' = X\mu + \mu', \ \mathbf{l}'' = X\mathbf{l} + \mathbf{l}', \ \mathbf{r}'' = X\mathbf{r} + \mathbf{r}'.$$

$$y'' = Xy + y', \ z'' = Xz + z', \text{ and } x'' = Xx + x'.$$

By applying Equation (5) with these inputs, we derive the full list of coefficients $f_{d,t}^{\mathcal{V}^{comm}}$, for $0 \le d \le n+Q$ and $0 \le t \le d$. First, we define the common map:

$$g_{d,t} := \sum_{j=1}^{n}\sum_{i=1}^{Q}(w_{L_{i,j}} \sum_{\substack{w\in\{0,1\}\\ u\in[0,j-1]\\ v\in[0,i]\\ u+v+w=t}} \binom{j-1}{u}\binom{i}{v}x^w(x')^{1-w}y^u(y')^{j-1-u}z^v(z')^{i-v})H_j$$

$$+\sum_{j=1}^{n}\sum_{i=1}^{Q}(w_{R_{i,j}}\sum_{\substack{w\in\{0,1\}\\u\in[0,j-1]\\v\in[0,i]\\u+v+w=t}}\binom{j-1}{u}\binom{i}{v}x^w(x')^{1-w}y^u(y')^{j-1-u}z^v(z')^{i-v})G_j$$

$$+\sum_{j=1}^{n}\sum_{i=1}^{Q}(w_{O_{i,j}}\sum_{\substack{u\in[0,j-1]\\v\in[0,i]\\u+v=t}}\binom{j-1}{u}\binom{i}{v}y^u(y')^{j-1-u}z^v(z')^{i-v})H_j$$

$$-\sum_{j=1}^{n}(\sum_{\substack{u\in[0,j-1]\\w\in\{0,1\}\\u+w=t}}\binom{j-1}{u}y^u(y')^{j-1-u}r_j^w(r'_j)^{1-w})H_j,$$

with $w_{L_{i,j}} = w_{R_{i,j}} = 0$ if $i+j \neq d$, $w_{O_{i,j}} = 0$ if $i+j-1 \neq d$ and $r_j = 0$ if $j \neq d$. Then, we have the full decomposition:

$$f_{0,0}^{\mathcal{V}^{comm}} = -\langle \mathbf{1}^n, \mathbf{H}\rangle,$$

$$f_{1,0}^{\mathcal{V}^{comm}} = -\mu' H - \langle \mathbf{l}', \mathbf{G}\rangle + g_{1,0},\ f_{1,1}^{\mathcal{V}^{comm}} = -\mu H - \langle \mathbf{l}, \mathbf{G}\rangle + g_{1,1},$$

$$f_{2,0}^{\mathcal{V}^{comm}} = x'A'_I + g_{2,0},\ f_{2,1}^{\mathcal{V}^{comm}} = xA'_I + x'A_I + g_{2,1},\ f_{2,2}^{\mathcal{V}^{comm}} = xA_I + g_{2,2},$$

$$f_{3,t}^{\mathcal{V}^{comm}} = \binom{2}{t}x^t(x')^{2-t}A'_O + \binom{2}{t-1}x^{t-1}(x')^{3-t}A_O + g_{3,t},$$

$$f_{4,t}^{\mathcal{V}^{comm}} = \binom{3}{t}x^t(x')^{3-t}S' + \binom{3}{t-1}x^{t-1}(x')^{4-t}S + g_{4,t},$$

$$f_{(4<d\leq n),t}^{\mathcal{V}^{comm}} = g_{d,t}.$$

and $f_{d,t}^{\mathcal{V}^{comm}}$ for $n < d \leq n+Q$ follow from the previous but without the folded terms of $-\langle \mathbf{y}^n \circ \mathbf{r}, \mathbf{H}\rangle$.

Finally, consider Equation (9) and let

$$\hat{t}''(X) = X\hat{t} + \hat{t}',\ \mathbf{l}''(X) = X\mathbf{l} + \mathbf{l}',\ \mathbf{r}''(X) = X\mathbf{r} + \mathbf{r}'.$$

By applying Equation (5) with these inputs, we derive the full list of coefficients $f_{d,t}^{\hat{t}}$, for $0 \leq d \leq 2$ and $0 \leq t \leq d$:

$$f_{0,0}^{\hat{t}} = 0$$

$$f_{1,0}^{\hat{t}} = \hat{t}',\ f_{1,1}^{\hat{t}} = \hat{t},$$

$$f_{2,0}^{\hat{t}} = -\langle \mathbf{l}', \mathbf{r}'\rangle,\ f_{2,1}^{\hat{t}} = -\langle \mathbf{l}, \mathbf{r}'\rangle - \langle \mathbf{l}', \mathbf{r}\rangle,\ f_{2,2}^{\hat{t}} = -\langle \mathbf{l}, \mathbf{r}\rangle.$$

5 Bulletproofs∗ protocols

5.1 Commit and Open arithmetic circuits

We transform the $(2k+1)$-move special-sound modified arithmetic circuit proof $(\mathcal{P}, \mathcal{V})$, presented in Figure 1, into a $(2k+1)$-move special-sound Commit and Open arithmetic circuit proof $(\mathcal{P}_{cm}, \mathcal{V}_{cm})$, for the following relation:

$$\mathcal{R}_{CM} = \{(\text{pi}; w) : (\text{pi}; w) \in \mathcal{R}_{AC}\}, \tag{10}$$

where statement pi and witness w are the same of Relation (2). The prover $\mathcal{P}_{cm}$ runs $\mathcal{P}$ to generate the prover messages $(\mathbf{m_i})_{i=1}^{k+1}$ of the argument $(\mathcal{P}, \mathcal{V})$. After receiving the final verifier challenge, $\mathcal{P}_{cm}$ computes and commits to the final messages $\mathbf{m}_{k+1}$, and sends both the commitment and its openings. The openings are defined by the tuple $\mathbf{m}_{k+1} := (\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r})$, i.e., the final prover messages in the argument $(\mathcal{P}, \mathcal{V})$. The verifier $\mathcal{V}_{cm}$ checks the correctness of the commitment and runs $\mathcal{V}$ on the public statement, prover messages, opening messages and public-coin challenges. The interactive argument $(\mathcal{P}_{cm}, \mathcal{V}_{cm})$ is shown in Figure 2.

$(\mathcal{P}_{cm}, \mathcal{V}_{cm})$'s **input**: $(\text{pi}; w) \in \mathcal{R}_{AC}$

$\mathcal{V}_{cm}$'s **output**: $\{0, 1\}$

$\mathcal{P}_{cm}$:

  $A_I, A_O, S \leftarrow \mathcal{P}(\text{pi}, w)$

$\mathcal{P}_{cm} \rightarrow \mathcal{V}_{cm} : A_I, A_O, S$

$\mathcal{V}_{cm} \rightarrow \mathcal{P}_{cm} : y, z \xleftarrow{\$} \mathbb{Z}_p$

$\mathcal{P}_{cm}$:

  $T_{1,3,4,5,6} \leftarrow \mathcal{P}(\text{pi}, w, y, z)$

$\mathcal{P}_{cm} \rightarrow \mathcal{V}_{cm} : T_1, T_3, T_4, T_5, T_6$

$\mathcal{V}_{cm} \rightarrow \mathcal{P}_{cm} : x \xleftarrow{\$} \mathbb{Z}_p$

$\mathcal{P}_{cm}$:

  $(\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}) \leftarrow \mathcal{P}(\text{pi}, w, y, z, x)$

  $C_{k+1} \leftarrow Commit(\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r})$

$\mathcal{P}_{cm} \rightarrow \mathcal{V}_{cm} : \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}, C_{k+1}$

$\mathcal{V}_{cm}$:

  $Commit(\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}) \stackrel{?}{=} C_{k+1}$

  $\mathcal{V}(\text{pi}, A_I, A_O, S, T_1, T_3, T_4, T_5, T_6, \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}, y, z, x) \stackrel{?}{=} 1$

Figure 2: Commit and Open arithmetic circuit interactive argument $(\mathcal{P}_{cm}, \mathcal{V}_{cm})$.

Lemma 2 ($(\mathcal{P}_{cm}, \mathcal{V}_{cm})$ is $(n(Q+1), 7)$-special-sound). *Let $(\mathcal{P}, \mathcal{V})$ be the $(n(Q+1), 7)$-special-sound argument for relation $\mathcal{R}_{AC}$ by Theorem 1. Then, the Commit and Open argument $(\mathcal{P}_{cm}, \mathcal{V}_{cm})$ is $(n(Q+1), 7)$-special-sound argument for the same relation $\mathcal{R}_{AC}$.*

Proof. Let $\mathcal{E}$ be the extractor for $(\mathcal{P}, \mathcal{V})$. We construct an extractor $\mathcal{E}'$ for $(\mathcal{P}_{cm}, \mathcal{V}_{cm})$. Given an $(n(Q+1), 7)$-tree of accepting transcripts for $(\mathcal{P}_{cm}, \mathcal{V}_{cm})$, the extractor $\mathcal{E}'$ removes the commitment C_{k+1} from every transcript. Since each transcript is accepting, the result is a valid $(n(Q+1), 7)$-tree of transcript for $(\mathcal{P}, \mathcal{V})$. Therefore, the extractor $\mathcal{E}'$ invokes $\mathcal{E}$ on this tree and returns its output, namely either a valid witness w such that $w \in \mathcal{R}_{AC}(pi)$ or a non-trivial DLOG relation. This concludes the proof.

Note that the binding property of the commitment scheme is not required to establish the special-soundness stated in Lemma 2. This is because our Commit and Open transform only commits to the final prover response, while the previous prover messages are kept explicitly. However, the binding property is essential for the knowledge-soundness of the folding scheme presented in Section 5.2.

5.2 Folding scheme for arithmetic circuits

We present a folding scheme $(\mathcal{P}_{fold}, \mathcal{V}_{fold}, \mathcal{D})$ for Bulletproofs, which we call *Bulletproofs∗*. To this end, we consider the NARK of the Commit and Open arithmetic circuit proof presented in Figure 2, which adopts the algebraic verifiers $\mathcal{V}^{\delta_{\text{new}}}$, $\mathcal{V}^{comm}$ and $\mathcal{V}^{\hat{t}}$ introduced in Section 4.2. We instantiate two random oracles, $\mathcal{O}_{NARK}$ and $\mathcal{O}_{fold}$, for our scheme. All algorithms share the CRS and circuit description parameters from Relation (2). For ease of exposition, we treat these parameters as implicit inputs to the folding scheme.

Folding prover $\mathcal{P}_{fold}$. On input the statement $\text{pi} = \{\mathbf{V}, \mathbf{c}\}$, the NARK proof π, and the accumulator acc, where

$$\pi = \{\pi_x = \{A_I, A_O, S, T_1, T_3, \ldots, T_6, C_{k+1}\}, \pi_w = \{\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\}\},$$

$$\text{acc} = \{\text{acc}_x = \{\mathbf{V}', \mathbf{c}', A_I', A_O', S', T_1', T_3', \ldots, T_6', C_{k+1}', y', z', x', E^{(1)}, E^{(2)}, E^{(3)}, \sigma\},$$
$$\text{acc}_w = \{\tau_x', \mu', \hat{t}', \mathbf{l}', \mathbf{r}'\}\},$$

the folding prover $\mathcal{P}_{fold}$ works as in Figure 3.

Folding verifier $\mathcal{V}_{fold}$. On input the statement pi, the NARK proof instance π_x, the accumulator instance acc_x, the updated accumulator instance

$$\text{acc}_x' = \{\mathbf{V}'', \mathbf{c}'', A_I'', A_O'', S'', T_1'', T_3'', \ldots, T_6'', C_{k+1}'', y'', z'', x'', E'^{(1)}, E'^{(2)}, E'^{(3)}, \sigma'\},$$

and the accumulation proof pf, the folding verifier $\mathcal{V}_{fold}$ works as in Figure 4.

Decider $\mathcal{D}$. On input an accumulator

$$\text{acc} = \{\text{acc}_x = \{\mathbf{V}, \mathbf{c}, A_I, A_O, S, T_1, T_3, \ldots, T_6, C_{k+1}, y, z, x, E^{(1)}, E^{(2)}, E^{(3)}, \sigma\},$$
$$\text{acc}_w = \{\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\}\},$$

the decider $\mathcal{D}$ works as in Figure 5.

input: $(\text{pi}, \pi, \text{acc})$

output: $\{\text{acc}', \text{pf}\}$

$y \leftarrow \mathcal{O}_{NARK}(V_1||\ldots||V_m||c_1||\ldots||c_Q||A_I||A_O||S)$

$z \leftarrow \mathcal{O}_{NARK}(y)$

$x \leftarrow \mathcal{O}_{NARK}(z||T_1||T_3||T_4||T_5||T_6)$

$\mathbf{V}''(X) = (V_j''(X))_{j=1}^m = (XV_j + V_j')_{j=1}^m$

$\mathbf{c}''(X) = (c_i''(X))_{i=1}^Q = (Xc_i + c_i')_{i=1}^Q$

$A_I''(X) = XA_I + A_I',\ A_O''(X) = XA_O + A_O',\ S''(X) = XS + S'$

$T_1''(X) = XT_1 + T_1',\ T_3''(X) = XT_3 + T_3', \ldots, T_6''(X) = XT_6 + T_6'$

$\tau_x''(X) = X\tau_x + \tau_x',\ \mu''(X) = X\mu + \mu',\ \hat{t}''(X) = X\hat{t} + \hat{t}'$

$\mathbf{l}''(X) = X\mathbf{l} + \mathbf{l}',\ \mathbf{r}''(X) = X\mathbf{r} + \mathbf{r}',\ C_{k+1}''(X) = XC_{k+1} + C_{k+1}'$

$y''(X) = Xy + y',\ z''(X) = Xz + z',\ x''(X) = Xx + x'$

$e^{(1)}(X) = \mathbf{e}^{(1)} + \sum_{j=1}^{D_1-1} \mathbf{e}_j^{(1)} X^j$ // $\sum_{d=0}^{D_1} (X+\sigma)^{D_1-d} \cdot f_d^{\mathcal{V}^{\delta_{\text{new}}}}(u''(X))$

$e^{(2)}(X) = \mathbf{e}^{(2)} + \sum_{j=1}^{D_2-1} \mathbf{e}_j^{(2)} X^j$ // $\sum_{d=0}^{D_2} (X+\sigma)^{D_2-d} \cdot f_d^{\mathcal{V}^{comm}}(u''(X))$

$e^{(3)}(X) = \mathbf{e}^{(3)} + \mathbf{e}_1^{(3)} X$ // $\sum_{d=0}^{2} (X+\sigma)^{2-d} \cdot f_d^{\mathcal{V}^{\hat{t}}}(u''(X))$

$\alpha \leftarrow \mathcal{O}_{fold}(\text{acc}_x||\text{pi}||\pi_x||\mathbf{e}_1^{(1)}||\ldots||\mathbf{e}_{D_1-1}^{(1)}||\mathbf{e}_1^{(2)}||\ldots||\mathbf{e}_{D_2-1}^{(2)}||\mathbf{e}_1^{(3)})$

$E'^{(1)} = E^{(1)} + e^{(1)}(\alpha) - e^{(1)}(0)$

$E'^{(2)} = E^{(2)} + e^{(2)}(\alpha) - e^{(2)}(0)$

$E'^{(3)} = E^{(3)} + e^{(3)}(\alpha) - e^{(3)}(0)$

$acc' = \{\mathbf{V}''(\alpha), \mathbf{c}''(\alpha), A_I''(\alpha), A_O''(\alpha), S''(\alpha), T_1''(\alpha), T_3''(\alpha), \ldots, T_6''(\alpha), C_{k+1}''(\alpha),$

$y''(\alpha), z''(\alpha), x''(\alpha), E'^{(1)}, E'^{(2)}, E'^{(3)}, \sigma' = \alpha + \sigma; \tau_x''(\alpha), \mu''(\alpha), \hat{t}''(\alpha), \mathbf{l}''(\alpha),$

$\mathbf{r}''(\alpha)\}$

$\text{pf} = \{\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_{D_1-1}^{(1)}, \mathbf{e}_1^{(2)}, \ldots, \mathbf{e}_{D_2-1}^{(2)}, \mathbf{e}_1^{(3)}\}$

Figure 3: Folding Prover $\mathcal{P}_{\text{fold}}^{\mathcal{O}_{NARK}, \mathcal{O}_{fold}}$.

input: $(\text{pi}, \pi_x, \text{acc}_x, \text{acc}_x', \text{pf})$

output: $\{0, 1\}$

$y \leftarrow \mathcal{O}_{NARK}(V_1||\ldots||V_m||c_1||\ldots||c_Q||A_I||A_O||S)$

$z \leftarrow \mathcal{O}_{NARK}(y)$

$x \leftarrow \mathcal{O}_{NARK}(z||T_1||T_3||T_4||T_5||T_6)$

$\alpha \leftarrow \mathcal{O}_{fold}(\text{acc}_x||\text{pi}||\pi_x||\mathbf{e}_1^{(1)}||\ldots||\mathbf{e}_{D_1-1}^{(1)}||\mathbf{e}_1^{(2)}||\ldots||\mathbf{e}_{D_2-1}^{(2)}||\mathbf{e}_1^{(3)})$

$(V_j'')_{j=1}^m \stackrel{?}{=} (\alpha \cdot V_j + V_j')_{j=1}^m,\ (c_i'')_{i=1}^Q \stackrel{?}{=} (\alpha \cdot c_i + c_i')_{i=1}^Q$

$y'' \stackrel{?}{=} \alpha \cdot y + y',\ z'' \stackrel{?}{=} \alpha \cdot z + z',\ x'' \stackrel{?}{=} \alpha \cdot x + x'$

$A_I'' \stackrel{?}{=} \alpha \cdot A_I + A_I',\ A_O'' \stackrel{?}{=} \alpha \cdot A_O + A_O',\ S'' \stackrel{?}{=} \alpha \cdot S + S'$

$T_1'' \stackrel{?}{=} \alpha \cdot T_1 + T_1',\ (T_i'')_{i=3}^6 \stackrel{?}{=} (\alpha \cdot T_i + T_i')_{i=3}^6$

11: $C''_{k+1} \stackrel{?}{=} \alpha \cdot C_{k+1} + C'_{k+1},\ \sigma' \stackrel{?}{=} \alpha + \sigma$
12: $E'^{(1)} \stackrel{?}{=} E^{(1)} + \sum_{j=1}^{D_1-1} \alpha^j \cdot \mathbf{e}_j^{(1)}$
13: $E'^{(2)} \stackrel{?}{=} E^{(2)} + \sum_{j=1}^{D_2-1} \alpha^j \cdot \mathbf{e}_j^{(2)}$
14: $E'^{(3)} \stackrel{?}{=} E^{(3)} + \alpha \cdot \mathbf{e}_1^{(3)}$

Figure 4: Folding Verifier $\mathcal{V}_{\text{fold}}^{\mathcal{O}_{NARK},\mathcal{O}_{fold}}$.

1: **input**: (acc)
2: **output**: $\{0,1\}$
3: $C_{k+1} \stackrel{?}{=} Commit(\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r})$
4: $E^{(1)} \stackrel{?}{=} \sum_{d=0}^{D_1} \sigma^{D_1-d} \cdot f_d^{\mathcal{V}^{\delta_{\text{new}}}}(\mathbf{V}, \mathbf{c}, T_1, T_3, \ldots, T_6, \tau_x, \hat{t}, y, z, x)$ // $D_1 = n + 1 + 2Q$
5: $E^{(2)} \stackrel{?}{=} \sum_{d=0}^{D_2} \sigma^{D_2-d} \cdot f_d^{\mathcal{V}^{comm}}(A_I, A_O, S, \mu, \mathbf{l}, \mathbf{r}, y, z, x)$ // $D_2 = n + Q$
6: $E^{(3)} \stackrel{?}{=} \sum_{d=0}^{2} \sigma^{2-d} \cdot f_d^{\mathcal{V}^{\hat{t}}}(\hat{t}, \mathbf{l}, \mathbf{r})$

Figure 5: Decider $\mathcal{D}$.

The following theorem shows that the folding scheme $(\mathcal{P}_{fold}, \mathcal{V}_{fold}, \mathcal{D})$ is secure.

Theorem 2 (Bulletproofs∗ folding scheme). *The folding scheme $(\mathcal{P}_{fold}, \mathcal{V}_{fold}, \mathcal{D})$ has perfect completeness and knowledge soundness.*

Proof sketch. Completeness and knowledge soundness follow the high-level argument of ProtoStar, with specific adaptations due to the underlying Bulletproofs∗ protocols. In order to prove completeness, let $(\text{pi}, \pi, \text{acc}) \in \mathcal{R}_{fold}$, that is, $\mathcal{V}_{NARK}(\text{pi}, \pi) = 1$ and $\mathcal{D}(acc) = 1$. Then, $\mathcal{D}(\text{acc}') = 1$ and $\mathcal{V}_{fold}(\text{pi}, \pi_x, \text{acc}_x, \text{acc}'_x, \text{pf}) = 1$, where (acc', pf) is the output of $\mathcal{P}_{fold}(\text{pi}, \pi, \text{acc})$. This holds if the linear combinations through α are consistent in $\mathcal{P}_{fold}$ and $\mathcal{V}_{fold}$, the commitment scheme is homomorphic, and the accumulation proof is computed correctly. Knowledge soundness follows by the existence of a special-sound (public-coin) interactive protocol and by applying the Fiat-Shamir transform to it. The interactive protocol $(\mathcal{P}, \mathcal{V})$ is defined as follows. $\mathcal{P}$ sends the accumulation proof pf, $\mathcal{V}$ replies with a random challenge α, then $\mathcal{P}$ responds with acc'. The verifier $\mathcal{V}$ accepts if $\mathcal{D}(\text{acc}') = 1$ and $\mathcal{V}_{fold}(\text{pi}, \pi_x, \text{acc}_x, \text{acc}'_x, \text{pf}) = 1$ using the random challenge α. This concludes the proof sketch.

The complete proof appears in Appendix B.

5.3 Complexity of the folding scheme

We provide the computational complexity of the folding scheme $(\mathcal{P}_{fold}, \mathcal{V}_{fold}, \mathcal{D})$; the dimensions n, m and Q are specified in Relation (2), and MSM denotes a multi-scalar multiplication.

The folding prover $\mathcal{P}_{fold}$:

- does 3 queries to $\mathcal{O}_{NARK}$ and 1 query to $\mathcal{O}_{fold}$;
- computes the coefficients of a degree $D_1 = n + 1 + 2Q$ polynomial for $\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_{D_1-1}^{(1)}$, the coefficients of a degree $D_2 = n + Q$ polynomial for $\mathbf{e}_1^{(2)}, \ldots, \mathbf{e}_{D_2-1}^{(2)}$, and the coefficient of a degree 2 polynomial for $\mathbf{e}_1^{(3)}$;
- 1 MSM of size $D_1 - 1 = n + 2Q$ plus 1 group addition to compute $E'^{(1)}$;
- 1 MSM of size $D_2 - 1 = n + Q - 1$ plus 1 group addition to compute $E'^{(2)}$;
- 2 field operations to compute $E'^{(3)}$;
- $m + 9$ scalar multiplications plus $m + 9$ group additions to compute $\mathbf{V}'', A_I'', A_O'', S'', T_1'', T_3'', \ldots, T_6'', C_{k+1}''$;
- $2Q + 4n + 13$ field operations to compute $\mathbf{c}'', y'', z'', x'', \sigma', \tau_x'', \mu'', \hat{t}'', \mathbf{l}'', \mathbf{r}''$.

The folding verifier $\mathcal{V}_{fold}$:

- does 3 queries to $\mathcal{O}_{NARK}$ and 1 query to $\mathcal{O}_{fold}$;
- $m + 9$ scalar multiplications plus $m + 9$ group additions to check $\mathbf{V}'', A_I'', A_O'', S'', T_1'', T_3'', \ldots, T_6'', C_{k+1}''$;
- $2Q + 7$ field operations to check $\mathbf{c}'', y'', z'', x'', \sigma'$;
- 1 MSM of size $D_1 - 1 = n + 2Q$ plus 1 group addition to check $E'^{(1)}$;
- 1 MSM of size $D_2 - 1 = n + Q - 1$ plus 1 group addition to check $E'^{(2)}$;
- 2 field operations to check $E'^{(3)}$.

The decider $\mathcal{D}$:

- checks the consistency of commitment C_{k+1} with 2 MSMs of size n plus 3 scalar multiplications plus 4 group additions;
- checks $E^{(1)}$ by evaluating the relaxed verifier $V^{\delta_{new}}$ of degree $D_1 = n+1+2Q$;
- checks $E^{(2)}$ by evaluating the relaxed verifier V^{comm} of degree $D_2 = n + Q$;
- checks $E^{(3)}$ by evaluating the relaxed verifier $V^{\hat{t}}$ of degree 2.

6 Comparison

Following Theorem 3 of ProtoStar, we can evaluate the performance of our folding scheme after the folding-to-IVC transformation. To show that this compilation is non-trivial, we estimate the runtime of the resulting IVC verifier and compare it with the cost of the original monolithic Bulletproofs verifier over N invocations. Consider the interactive protocol of the modified Bulletproofs in Figure 1, i.e., a 5-move special-sound interactive protocol with $k + 1$ prover message rounds and $k = 2$ verifier challenge rounds. Moreover, the interactive verifier checks three algebraic maps of output length 1 each. Then, specializing to our construction, there exists an IVC scheme $(\mathcal{P}_{IVC}, \mathcal{V}_{IVC})$ and $\mathcal{V}_{IVC}$ has the following efficiency:

$$3 + |m_{k+1}|\mathbb{G} + \mathcal{V}_{BP*}, \tag{11}$$

where the quantity 3 is related to the total output length l of the algebraic maps, $|m_{k+1}|$ is the prover message length, $\mathbb{G}$ is the total length of the messages

committed using $Commit$, and $\mathcal{V}_{BP*}$ the cost of running the algebraic verifier of the interactive protocol. The quantity in (11) compares with the cost of verifying N proofs via the monolithic Bulletproofs verifier, which is approximately $N \cdot \mathcal{V}_{BP}$. Hence, verifying folded proofs is asymptotically more efficient than verifying N proofs via a monolithic Bulletproofs verifier, with a gain that scales linearly in N.

7 Conclusions and future directions

In this work, we presented Bulletproofs*, a folding scheme for arithmetic circuit proofs under standard assumptions and without preprocessing. To this end, we first adapted the algebraic verifiers of the arithmetic circuit proof of Bulletproofs to the algebraic form required by ProtoStar, and proved that the modified protocol remains secure. Then, we designed the Bulletproofs folding scheme that is complete and knowledge-sound. Finally, we analyzed the resulting verifier cost after the folding-to-IVC transformation. The result shows an asymptotic linear gain compared to repeated invocations of the monolithic Bulletproofs verifier.
In the current design, the recursive circuit is linear in the degree of the algebraic verifier, and hence the arithmetic circuit complexity. This may be optimized by reducing the number of group operations using the ProtoStar compression technique, i.e., by compressing verifier checks via a random linear combination. Moreover, assuming constant-sized statements, the accumulator witness size is also linear in the circuit complexity; introducing an IPA could reduce it to logarithmic size. We leave these optimizations to future work, together with an implementation for concrete performance evaluation and an extended comparison.

References

1. Alonso, K.M., et al.: Zero to monero (2020)
2. Attema, T., Cramer, R.: Compressed-protocol theory and practical application to plug & play secure algorithmics. In: Annual International Cryptology Conference. pp. 513–543. Springer (2020)
3. Attema, T., Fehr, S., Klooß, M.: Fiat-shamir transformation of multi-round interactive proofs. In: Theory of Cryptography Conference. pp. 113–142. Springer (2022)
4. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 62–73 (1993)
5. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Scalable zero knowledge via cycles of elliptic curves. Algorithmica **79**(4), 1102–1160 (2017)
6. Boneh, D., Drake, J., Fisch, B., Gabizon, A.: Halo infinite: Proof-carrying data from additive polynomial commitments. In: Annual International Cryptology Conference. pp. 649–680. Springer (2021)
7. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 327–357. Springer (2016)

8. Bowe, S., Grigg, J., Hopwood, D.: Recursive proof composition without a trusted setup. Cryptology ePrint Archive (2019)
9. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: Towards privacy in a smart contract world. In: International Conference on Financial Cryptography and Data Security. pp. 423–443. Springer (2020)
10. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE symposium on security and privacy (SP). pp. 315–334. IEEE (2018)
11. Bünz, B., Chen, B.: Protostar: Generic efficient accumulation/folding for special-sound protocols. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 77–110. Springer (2023)
12. Bünz, B., Chiesa, A., Lin, W., Mishra, P., Spooner, N.: Proof-carrying data without succinct arguments. In: Annual International Cryptology Conference. pp. 681–710. Springer (2021)
13. Bünz, B., Chiesa, A., Mishra, P., Spooner, N.: Proof-carrying data from accumulation schemes. Cryptology ePrint Archive (2020)
14. Chung, H., Han, K., Ju, C., Kim, M., Seo, J.H.: Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger. Ieee Access **10**, 42081–42096 (2022)
15. Eagen, L., Kanjalkar, S., Ruffing, T., Nick, J.: Bulletproofs++: next generation confidential transactions via reciprocal set membership arguments. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 249–279. Springer (2024)
16. Fauzi, P., Meiklejohn, S., Mercer, R., Orlandi, C.: Quisquis: A new design for anonymous cryptocurrencies. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 649–678. Springer (2019)
17. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the theory and application of cryptographic techniques. pp. 186–194. Springer (1986)
18. Goodell, B.: Generalized bulletproofs for opening vector commitments. `https://github.com/cypherstack/generalized-bulletproofs-fix/blob/main/generalized_bulletproofs_fix.pdf` (2026), accessed: 2026-05-18
19. Kothapalli, A., Setty, S., Tzialla, I.: Nova: Recursive zero-knowledge arguments from folding schemes. In: Annual International Cryptology Conference. pp. 359–388. Springer (2022)
20. Lindell: Parallel coin-tossing and constant-round secure two-party computation. Journal of Cryptology **16**, 143–184 (2003)
21. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. pp. 2111–2128 (2019)
22. Scala, E.: Trustless zero-knowledge proofs in public blockchain (2024)
23. Scala, E., Dong, C., Corradini, F., Mostarda, L.: Zeromt: Towards multi-transfer transactions with privacy for account-based blockchain. Journal of Information Security and Applications **83**, 103794 (2024)
24. Scala, E., Mostarda, L.: Efficient inner-product argument from compressed σ-protocols and applications. In: International Conference on Advanced Information Networking and Applications. pp. 326–337. Springer (2024)
25. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. Journal of the ACM (JACM) **27**(4), 701–717 (1980)
26. Thaler, J.: Proofs, arguments, and zero-knowledge. Foundations and Trends® in Privacy and Security **4**(2-4), 117–660 (2022)

A Proof of Theorem 1

We prove perfect completeness, perfect honest-verifier zero-knowledge and special-soundness for the 5-move public-coin argument in Figure 1.
For perfect completeness, if the prover knows a valid witness and adheres to the protocol, it must hold that

$$t_2 = \langle \mathbf{a}_L, \mathbf{y}^{-n} \circ \mathbf{a}_R \rangle - \langle \mathbf{a}_O, \mathbf{y}^{-n} \rangle$$

$$+\langle \mathbf{a}_L, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L \rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R, \mathbf{a}_R \rangle + \langle \mathbf{a}_O, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O \rangle$$

$$+\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L \rangle$$

$$= \langle \mathbf{W}_L \cdot \mathbf{a}_L + \mathbf{W}_R \cdot \mathbf{a}_R + \mathbf{W}_O \cdot \mathbf{a}_O, \mathbf{z}_{[1:]}^{Q+1} \rangle + \delta_{new}(y, z)$$

$$= \langle \mathbf{c} + \mathbf{W}_V \cdot \mathbf{v}, \mathbf{z}_{[1:]}^{Q+1} \rangle + \delta_{new}(y, z).$$

To prove perfect honest-verifier zero-knowledge, we construct an efficient a simulator that, given the statement and the public-coin randomness, outputs a transcript that is identically distributed to the transcript produced in an honest execution between prover and verifier. Given the statement and the verifier's randomness x, y, z, the simulator computes:

$$\mu, \tau_x \xleftarrow{\$} \mathbb{Z}_p$$

$$\mathbf{l}, \mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^n$$

$$\hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle$$

$$A_I, A_O \xleftarrow{\$} \mathbb{G}$$

$$S = x^{-3}(\mu H + \langle \mathbf{l}, \mathbf{G} \rangle + \langle \mathbf{y}^n \circ \mathbf{r}, \mathbf{H} \rangle - xA_I - x^2 A_O + \langle \mathbf{1}^n, \mathbf{H} \rangle$$

$$-x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L), \mathbf{H} \rangle - x\langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{G} \rangle - \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O), \mathbf{H} \rangle)$$

$$T_{3,4,5,6} \xleftarrow{\$} \mathbb{G}$$

$$T_1 = x^{-1}(\hat{t}G + \tau_x H - x^2 \delta_{\text{new}}(y, z)G - x^2 \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle G - x^2 \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle - \sum_{i=3}^{6} x^i T_i)$$

Hence, the simulator outputs the transcript:

$$(A_I, A_O, S; y, z; T_1, T_3, \ldots, T_6; x; \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r})$$

This simulated transcript is indistinguishable from that produced during an honest execution where: $A_O, A_I, \mathbf{l}, \mathbf{r}, \mu, \tau_x$ are random independent elements if $\gamma_I, \gamma_O, \mathbf{S}_L, \mathbf{S}_R, \gamma_O, \rho_1, (\rho_i)_{i=3}^6, \boldsymbol{\gamma}$ are chosen independently and randomly; $\hat{t}$ is the inner product of $\mathbf{l}, \mathbf{r}$; S is defined by Equation (8); $T_1, (T_i)_{i=3}^6$ are random group elements if they are perfectly hiding commitments and are defined by Equation (7); finally, x, y, z are uniformly selected challenges.

To prove special-soundness, we construct an extractor that rewinds the prover with n different challenges y, $(Q+1)$ different challenges z and 7 different challenges x, thus getting in total $n \cdot (Q+1) \cdot 7$ accepting transcripts.
We consider 3 valid transcripts of the form

$$((A_I, A_O, S, y, z, T_1, T_3 \ldots, T_6), x_i, (\tau_{x_i}, \mu_i, \hat{t}_i, \mathbf{l}_i, \mathbf{r}_i)),$$

for $1 \leq i \leq 3$, with different challenges x_1, x_2, x_3 and the same challenges y and z. Assuming that x_1, x_2, x_3 are pairwise distinct, we find three values q_1, q_2, q_3 such that

$$\sum_{i=1}^{3} q_i x_i = 1 \ \wedge \ \sum_{i=1}^{3} q_i x_i^2 = \sum_{i=1}^{3} q_i x_i^3 = 0.$$

By linearly combining this choice of values with Equation (8) as follows

$$\sum_{i=1}^{3} q_i x_i A_I + \sum_{i=1}^{3} q_i x_i^2 A_O + \sum_{i=1}^{3} q_i x_i^3 S - \sum_{i=1}^{3} q_i \langle \mathbf{1}^n, \mathbf{H} \rangle$$

$$+ \sum_{i=1}^{3} q_i x_i \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L), \mathbf{H} \rangle$$

$$+ \sum_{i=1}^{3} q_i x_i \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{G} \rangle$$

$$+ \sum_{i=1}^{3} q_i \langle \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O), \mathbf{H} \rangle =$$

$$\sum_{i=1}^{3} q_i \mu_i H + \sum_{i=1}^{3} q_i \langle \mathbf{l}_i, \mathbf{G} \rangle + \sum_{i=1}^{3} q_i \langle \mathbf{y}^n \circ \mathbf{r}_i, \mathbf{H} \rangle,$$

we compute

$$\hat{\mathbf{a}}_L = \sum_{i=1}^{3} q_i \mathbf{l}_i - \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R)$$

$$\hat{\mathbf{a}}_R = \mathbf{y}^n \circ (\sum_{i=1}^{3} q_i \mathbf{r}_i - (\sum_{i=1}^{3} q_i)(\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O) - (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L)) + (\sum_{i=1}^{3} q_i)\mathbf{1}^n$$

$$\hat{\gamma}_I = \sum_{i=1}^{3} q_i \mu_i$$

such that $A_I = \hat{\gamma}_I H + \langle \hat{\mathbf{a}}_L, \mathbf{G} \rangle + \langle \hat{\mathbf{a}}_R, \mathbf{H} \rangle$. For a different set of challenges x_1', x_2', x_3', unless we compute $\hat{\gamma}_I' = \hat{\gamma}_I$, $\hat{\mathbf{a}}_L' = \hat{\mathbf{a}}_L$, $\hat{\mathbf{a}}_R' = \hat{\mathbf{a}}_R$, different openings of A_I yield a non-trivial DLOG relation among H,$\mathbf{G}$,$\mathbf{H}$.

Using the same challenges, but combining another choice of values b_1, b_2, b_3 in Equation (8) as follows

$$\sum_{i=1}^{3} b_i x_i = 0 \ \wedge \ \sum_{i=1}^{3} b_i x_i^2 = 1 \ \wedge \sum_{i=1}^{3} b_i x_i^3 = 0,$$

we compute

$$\hat{\mathbf{a}}_{O,L} = \sum_{i=1}^{3} b_i \mathbf{l}_i$$

$$\hat{\mathbf{a}}_{O,R} = \mathbf{y}^n \circ (\sum_{i=1}^{3} b_i \mathbf{r}_i - (\sum_{i=1}^{3} b_i)(\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O)) + (\sum_{i=1}^{3} b_i)\mathbf{1}^n$$

$$\hat{\gamma_O} = \sum_{i=1}^{3} b_i \mu_i$$

such that $A_O = \hat{\gamma}_O H + \langle \hat{\mathbf{a}}_{O,L}, \mathbf{G} \rangle + \langle \hat{\mathbf{a}}_{O,R}, \mathbf{H} \rangle$. For a different set of challenges x_1', x_2', x_3', if we can compute different openings $\hat{\gamma}_O'$, $\hat{\mathbf{a}}_{O,L}'$, $\hat{\mathbf{a}}_{O,R}'$ for A_O, then we have found a non-trivial DLOG relation among H,$\mathbf{G}$,$\mathbf{H}$. Similarly, we can compute $\hat{\mathbf{s}}_L$, $\hat{\mathbf{s}}_R$, $\hat{\gamma_S}$ such that $S = \hat{\gamma}_S H + \langle \hat{\mathbf{s}}_L, \mathbf{G} \rangle + \langle \hat{\mathbf{s}}_R, \mathbf{H} \rangle$.
Considering the unique computed openings of A_I, A_O, S in Equation (8), we have that for $\mathbf{l}, \mathbf{r}, \hat{t}$ obtained from the transcripts and for all challenges x, y, z:

$$\mathbf{l} = (\hat{\mathbf{a}}_L + \mathbf{y}^n \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R))x + \hat{\mathbf{a}}_{O,L} x^2 + \hat{\mathbf{s}}_L x^3$$

$$\mathbf{r} = -\mathbf{y}^{-n} + \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O + (\mathbf{y}^{-n} \circ \hat{\mathbf{a}}_R + \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L)x + \mathbf{y}^{-n} \circ \hat{\mathbf{a}}_{O,R} x^2 + \mathbf{y}^{-n} \circ \hat{\mathbf{s}}_R x^3$$

$$\hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle$$

If these equalities do not hold for $\mathbf{l}, \mathbf{r}$ from the transcript and for all challenges, then

$$\langle \mathbf{l} - \hat{\mathbf{l}}, \mathbf{G} \rangle + \langle \mathbf{y}^n \circ (\mathbf{r} - \hat{\mathbf{r}}), \mathbf{H} \rangle = 0$$

with at least one non-zero coefficient of $\mathbf{G}$ and $\mathbf{H}$ and canceling the common term μH, where $\hat{\mathbf{l}}$ and $\hat{\mathbf{r}}$ are obtained from the computed openings. Hence, this means that we have found a non-trivial DLOG relation among H,$\mathbf{G}$,$\mathbf{H}$.
We now consider 6 valid transcripts of the form

$$((A_I, A_O, S, y, z, T_1, T_3 \ldots, T_6), x_i, (\tau_{x_i}, \mu_i, \hat{t}_i, \mathbf{l}_i, \mathbf{r}_i)),$$

for $1 \leq i \leq 6$, with different challenges $x_1, x_2, \ldots, x_6$ and fixed challenges y and z. Assuming that $x_1, x_2, \ldots, x_6$ are pairwise distinct, we find six values $o_1, o_2, \ldots, o_6$ such that

$$\sum_{i=1}^{6} o_i x_i = 1 \ \wedge \ \sum_{i=1}^{6} o_i x_i^2 = \sum_{i=1}^{6} o_i x_i^3 = \cdots = \sum_{i=1}^{6} o_i x_i^6 = 0,$$

and by linearly combining these values in Equation (7) as follows

$$\sum_{i=1}^{6} o_i \hat{t}_i G + \sum_{i=1}^{6} o_i \tau_{x_i} H =$$

$$\sum_{i=1}^{6} o_i x_i^2 \delta_{\text{new}}(y,z) G + \sum_{i=1}^{6} o_i x_i^2 \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle G$$

$$+ \sum_{i=1}^{6} o_i x_i^2 \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle + \sum_{i=1}^{6} o_i x_i T_1$$

$$+ \sum_{i=1}^{6} o_i x_i^3 T_3 + \cdots + \sum_{i=1}^{6} o_i x_i^6 T_6,$$

we compute

$$\rho_1 = \sum_{i=1}^{6} o_i \tau_{x_i}$$

$$t_1 = \sum_{i=1}^{6} o_i \hat{t}_i.$$

Similarly, for $k \in [2,6]$, we compute ρ_k, t_k by using the same challenges but a different set of six values for each k, chosen such that all terms except the x^k term cancel out in the linear combination. Note that $T_k = t_k G + \rho_k H$ for $k \in \{1,3,4,5,6\}$.

Using t_2, ρ_2 we can compute the aggregate openings v, γ such that $vG + \gamma H = \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle$. Indeed, from the linear combination that does not cancel out the quadratic terms, we have that

$$t_2 G + \rho_2 H = \delta_{new}(y,z) G + \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle G + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle$$

and hence

$$\langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle = (t_2 - \delta_{new}(y,z) - \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle) G + \rho_2 H$$

that implies

$$v = t_2 - \delta_{new}(y,z) - \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle$$

$$\gamma = \rho_2.$$

Using m different challenges z_s and linear combinations of $vG + \gamma H = \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle$, we can compute individual openings v_j, γ_j such that $v_j G + \gamma_j H = V_j$. Define $\mathbf{z}_{[1:],s}^{Q+1} = (z_s, z_s^2, \ldots, z_s^Q)$ for $s \in [1,m]$, $\mathbf{v} = (v_1, \ldots, v_m)$, and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)$. For fixed y we have

$$\langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \mathbf{V} \rangle =$$

$$\langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \mathbf{v} \rangle G + \langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \boldsymbol{\gamma} \rangle H =$$
$$(t_2^{(s)} - \delta_{new}(y, z_s) - \langle \mathbf{z}_{[1:],s}^{Q+1}, \mathbf{c} \rangle) G + \rho_2^{(s)} H.$$

Hence, we obtain

$$(t_2^{(s)} - \delta_{new}(y, z_s) - \langle \mathbf{z}_{[1:],s}^{Q+1}, \mathbf{c} \rangle - \langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \mathbf{v} \rangle) G - (\rho_2^{(s)} - \langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \boldsymbol{\gamma} \rangle) H = 0.$$

In this equality, both coefficients of G and H must be zero; otherwise, we obtain a non-trivial DLOG relation between G and H, similarly to the argument in [18]. Since this happens with negligible probability, this implies that

$$t_2^{(s)} = \delta_{new}(y, z_s) + \langle \mathbf{z}_{[1:],s}^{Q+1}, \mathbf{c} \rangle + \langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \mathbf{v} \rangle$$

$$\rho_2^{(s)} = \langle \mathbf{z}_{[1:],s}^{Q+1} \cdot \mathbf{W}_V, \boldsymbol{\gamma} \rangle.$$

Define the following matrix

$$M = \begin{pmatrix} \mathbf{z}_{[1:],1}^{Q+1} \\ \mathbf{z}_{[1:],2}^{Q+1} \\ \vdots \\ \mathbf{z}_{[1:],m}^{Q+1} \end{pmatrix} \mathbf{W}_V = \begin{pmatrix} z_1 & z_1^2 & \dots & z_1^Q \\ z_2 & z_2^2 & \dots & z_2^Q \\ \vdots & \vdots & & \vdots \\ z_m & z_m^2 & \dots & z_m^Q \end{pmatrix} \mathbf{W}_V \in \mathbb{Z}_p^{m \times m}. \tag{12}$$

If $\mathbf{W}_V$ has rank m, then necessarily $Q \geq m$, and M is invertible except with negligible probability. Indeed, the determinant of M is a non-zero multivariate polynomial in the random challenges $z_1, \dots, z_m$, and therefore it vanishes at the random values only with negligible probability by the Schwartz-Zippel lemma [25].
Hence, from

$$M \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} t_2^{(1)} - \delta_{new}(y, z_1) - \langle \mathbf{z}_{[1:],1}^{Q+1}, \mathbf{c} \rangle \\ \vdots \\ t_2^{(m)} - \delta_{new}(y, z_m) - \langle \mathbf{z}_{[1:],m}^{Q+1}, \mathbf{c} \rangle \end{pmatrix},$$

we can compute

$$v_j = \sum_{s=1}^{m} m_{j,s} \cdot (t_2^{(s)} - \delta_{new}(y, z_s) - \langle \mathbf{z}_{[1:],s}^{Q+1}, \mathbf{c} \rangle)$$

$$\gamma_j = \sum_{s=1}^{m} m_{j,s} \cdot \rho_2^{(s)},$$

where $m_{j,s}$ is the entry of M^{-1}.
Considering the unique computed openings of $T_1, T_3, \dots, T_6$ and $\mathbf{V}$ in Equation (7), we have that for all transcripts:

$$\hat{t} = t_1 x + (\delta_{new}(y, z) + \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{v} \rangle) x^2 + \sum_{i=3}^{6} t_i x^i$$

$$\tau_x = \rho_1 x + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \boldsymbol{\gamma} \rangle x^2 + \sum_{i=3}^{6} \rho_i x^i$$

If for any transcript these equalities do not hold, then

$$(\hat{t} - t_1 x - (\delta_{new}(y,z) + \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{v} \rangle) x^2 - \sum_{i=3}^{6} t_i x^i) G$$

$$+ (\tau_x - \rho_1 x - \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \boldsymbol{\gamma} \rangle x^2 - \sum_{i=3}^{6} \rho_i x^i) H = 0$$

with at least one non-zero coefficient of G and H. This implies that we have found a non-trivial DLOG relation among G, H.
Since this happens with negligible probability, this implies that

$$\hat{t} - \sum_{i=1}^{6} t_i x^i = 0$$

with $t_2 = \delta_{new}(y,z) + \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_V, \mathbf{v} \rangle$. Considering Equation (9) and substituting, we obtain that the degree-6 polynomial

$$\langle l(x), r(x) \rangle - \sum_{i=1}^{6} t_i x^i = 0, \tag{13}$$

vanishing at 7 distinct challenges $x = x_j$, for $j \in [1,7]$ and for all y, z challenges, is necessarily the zero polynomial, hence $\langle l(x), r(x) \rangle = \sum_{i=1}^{6} t_i x^i$.
Define $p(x) := \langle l(x), r(x) \rangle = \sum_{i=1}^{6} p_i \cdot x^i$, for any y, z, Equation (13) implies that $p_2 = t_2$, with

$$p_2 = \langle \hat{\mathbf{a}}_L, \mathbf{y}^{-n} \circ \hat{\mathbf{a}}_R \rangle - \langle \hat{\mathbf{a}}_{O,L}, \mathbf{y}^{-n} \rangle$$
$$+ \langle \hat{\mathbf{a}}_L, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L \rangle + \langle \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R, \hat{\mathbf{a}}_R \rangle + \langle \hat{\mathbf{a}}_{O,L}, \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O \rangle$$
$$+ \delta_{new}(y,z).$$

Hence, canceling out the term $\delta_{new}(y,z)$ from the equality $p_2 = t_2$, we obtain that the degree-Q polynomial (in z and with fixed y)

$$\langle \hat{\mathbf{a}}_L \circ \hat{\mathbf{a}}_R - \hat{\mathbf{a}}_{O,L}, \mathbf{y}^{-n} \rangle + \langle \mathbf{W}_L \cdot \hat{\mathbf{a}}_L + \mathbf{W}_R \cdot \hat{\mathbf{a}}_R + \mathbf{W}_O \cdot \hat{\mathbf{a}}_{O,L} - \mathbf{c} - \mathbf{W}_V \cdot \mathbf{v}, \mathbf{z}_{[1:]}^{Q+1} \rangle = 0$$

vanishing at $Q+1$ distinct challenges z, is necessarily the zero polynomial.
Thus, using $Q+1$ distinct challenges z and n distinct challenges y, i.e., $n(Q+1)$ transcripts in total, we can infer the following equalities:

$$\hat{\mathbf{a}}_L \circ \hat{\mathbf{a}}_R - \hat{\mathbf{a}}_{O,L} = \mathbf{0}^n \tag{14}$$

$$\mathbf{W}_L \cdot \hat{\mathbf{a}}_L + \mathbf{W}_R \cdot \hat{\mathbf{a}}_R + \mathbf{W}_O \cdot \hat{\mathbf{a}}_{O,L} = \mathbf{c} + \mathbf{W}_V \cdot \mathbf{v} \tag{15}$$

which are exactly the linear constraints on the arithmetic circuit relation. This means that we have extracted a valid witness $(\hat{\mathbf{a}}_L, \hat{\mathbf{a}}_R, \hat{\mathbf{a}}_O, \mathbf{v}, \boldsymbol{\gamma})$, where $\hat{\mathbf{a}}_O = \hat{\mathbf{a}}_{O,L}$, satisfying Relation (2). Finally, the extractor is efficient and processes a poly-bounded set of accepting transcripts. The extractor either extracts a valid witness or produces a non-trivial DLOG relation, which occurs only with negligible probability under DLOG assumption. Indeed, according to the generalization of Definition 4, the 5-move public-coin argument in Figure 1 is $(n(Q+1), 7)$-special-sound, where the extractor takes a $(n(Q+1), 7)$-tree of accepting transcripts.

B Proof of Theorem 2

We prove that the Bulletproofs∗ folding scheme $(\mathcal{P}_{fold}, \mathcal{V}_{fold}, \mathcal{D})$ presented in Section 5.2 has perfect completeness and knowledge soundness.
To prove completeness, consider $(\mathrm{pi}, \pi, \mathrm{acc}) \in \mathcal{R}_{fold}$, where

$$\mathrm{pi} = \{\mathbf{V}, \mathbf{c}\}$$

$$\pi = \{\pi_x = \{A_I, A_O, S, T_1, T_3, \ldots, T_6, C_{k+1}\}, \pi_w = \{\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\}\},$$

$$\mathrm{acc} = \{\mathrm{acc}_x = \{\mathbf{V}', \mathbf{c}', A_I', A_O', S', T_1', T_3', \ldots, T_6', C_{k+1}', y', z', x', E^{(1)}, E^{(2)}, E^{(3)}, \sigma\},$$

$$\mathrm{acc}_w = \{\tau_x', \mu', \hat{t}', \mathbf{l}', \mathbf{r}'\}\},$$

hence, $\mathcal{V}_{NARK}(\mathrm{pi}, \pi) = 1$ and $\mathcal{D}(acc) = 1$. Note that the CRS and circuit description parameters from Relation (2) are implicit for ease of exposition. Moreover, unlike Protostar, where the error terms are committed and the relative error commitment is derived, in our instantiation the errors are kept explicitly.
Consider the folding prover $\mathcal{P}_{fold}(\mathrm{pi}, \pi, \mathrm{acc})$ which outputs $(\mathrm{acc}', \mathrm{pf})$, where

$$\mathrm{acc}' = \{\mathrm{acc}_x' = \{\mathbf{V}'', \mathbf{c}'', A_I'', A_O'', S'', T_1'', T_3'', \ldots, T_6'', C_{k+1}'', y'', z'', x'',$$

$$E'^{(1)}, E'^{(2)}, E'^{(3)}, \sigma'\},$$

$$\mathrm{acc}_w' = \{\tau_x'', \mu'', \hat{t}'', \mathbf{l}'', \mathbf{r}''\}\},$$

$$\mathrm{pf} = \{\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_{D_1-1}^{(1)}, \mathbf{e}_1^{(2)}, \ldots, \mathbf{e}_{D_2-1}^{(2)}, \mathbf{e}_1^{(3)}\}.$$

Then $\mathcal{D}(\mathrm{acc}') = 1$ and $\mathcal{V}_{fold}(\mathrm{pi}, \pi_x, \mathrm{acc}_x, \mathrm{acc}_x', \mathrm{pf}) = 1$, that is, both will accept, thus proving perfect completeness. $\mathcal{V}_{fold}$ accepts since $\mathcal{P}_{fold}$ and $\mathcal{V}_{fold}$ compute y, z, x and α in the same way, thus the linear combinations of $(\mathrm{pi}, \pi_x, \mathrm{pf}, y, z, x)$ and acc_x through α are consistent.
We now show that $\mathcal{D}(\mathrm{acc}')$ accepts by inspecting the decider checks. In particular, the check $C_{k+1}'' \stackrel{?}{=} Commit(\tau_x'', \mu'', \hat{t}'', \mathbf{l}'', \mathbf{r}'')$ succeeds by the homomorphism of the commitment scheme. This is because

$$acc'.\{C_{k+1}'', \tau_x'', \mu'', \hat{t}'', \mathbf{l}'', \mathbf{r}''\} = \alpha \cdot \pi.\{C_{k+1}, \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\} + acc.\{C_{k+1}', \tau_x', \mu', \hat{t}', \mathbf{l}', \mathbf{r}'\}$$

where $\pi.C_{k+1} = Commit(\pi.\{\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\})$ because $\mathcal{V}_{NARK}(\mathrm{pi}, \pi)$ accepts, and $acc.C_{k+1}' = Commit(acc.\{\tau_x', \mu', \hat{t}', \mathbf{l}', \mathbf{r}'\})$ because $\mathcal{D}(\mathrm{acc})$ accepts.

Moreover, $\mathcal{D}(\mathrm{acc}')$ computes

$$E'^{(1)} = \sum_{d=0}^{D_1} (\sigma')^{D_1-d} \cdot f_d^{\mathcal{V}^{\delta_{\mathrm{new}}}}(\mathbf{V}'', \mathbf{c}'', T_1'', T_3'', \ldots, T_6'', \tau_x'', \hat{t}'', y'', z'', x''),$$

with $D_1 = n + 1 + 2Q$,

$$E'^{(2)} = \sum_{d=0}^{D_2} (\sigma')^{D_2-d} \cdot f_d^{\mathcal{V}^{comm}}(A_I'', A_O'', S'', \mu'', \mathbf{l}'', \mathbf{r}'', y'', z'', x'')$$

with $D_2 = n + Q$, and

$$E'^{(3)} = \sum_{d=0}^{2} (\sigma')^{2-d} \cdot f_d^{\mathcal{V}^{\hat{t}}}(\hat{t}'', \mathbf{l}'', \mathbf{r}''),$$

such that it holds

$$E'^{(1)} = E^{(1)} + \sum_{j=1}^{D_1-1} \alpha^j \cdot \mathrm{pf}.\mathbf{e}_j^{(1)} = \sum_{d=0}^{D_1} (\alpha + \sigma)^{D_1-d} \cdot f_d^{\mathcal{V}^{\delta_{\mathrm{new}}}}(u''(\alpha)), \tag{16}$$

$$E'^{(2)} = E^{(2)} + \sum_{j=1}^{D_2-1} \alpha^j \cdot \mathrm{pf}.\mathbf{e}_j^{(2)} = \sum_{d=0}^{D_2} (\alpha + \sigma)^{D_2-d} \cdot f_d^{\mathcal{V}^{comm}}(u''(\alpha)), \tag{17}$$

$$E'^{(3)} = E^{(3)} + \alpha \cdot \mathrm{pf}.\mathbf{e}_1^{(3)} = \sum_{d=0}^{2} (\alpha + \sigma)^{2-d} \cdot f_d^{\mathcal{V}^{\hat{t}}}(u''(\alpha)), \tag{18}$$

where

$$E^{(1)} = \sum_{d=0}^{D_1} \sigma^{D_1-d} \cdot f_d^{\mathcal{V}^{\delta_{\mathrm{new}}}}(\mathbf{V}', \mathbf{c}', T_1', T_3', \ldots, T_6', \tau_x', \hat{t}', y', z', x'),$$

$$E^{(2)} = \sum_{d=0}^{D_2} \sigma^{D_2-d} \cdot f_d^{\mathcal{V}^{comm}}(A_I', A_O', S', \mu', \mathbf{l}', \mathbf{r}', y', z', x'),$$

$$E^{(3)} = \sum_{d=0}^{2} \sigma^{2-d} \cdot f_d^{\mathcal{V}^{\hat{t}}}(\hat{t}', \mathbf{l}', \mathbf{r}'),$$

and $u''(\alpha)$ are the linear combinations

$$\alpha \cdot \{\mathrm{pi}.\{\mathbf{V}, \mathbf{c}\}, \pi.\{A_I, A_O, S, T_1, T_3, \ldots, T_6, \tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\}, y, z, x\}$$

$$+acc.\{\mathbf{V}', \mathbf{c}', A_I', A_O', S', T_1', T_3', \ldots, T_6', \tau_x', \mu', \hat{t}', \mathbf{l}', \mathbf{r}', y', z', x'\}.$$

Equation (16), Equation (17) and Equation (18) hold by definition of the intermediate-degree coefficients $\mathrm{pf}.\mathbf{e}_j^{(1)}$,$\mathrm{pf}.\mathbf{e}_j^{(2)}$,$\mathrm{pf}.\mathbf{e}_1^{(3)}$, since $\mathcal{D}(acc) = 1$ determines the constant coefficients $E^{(1)}, E^{(2)}, E^{(3)}$ and $\mathcal{V}_{NARK}(\mathrm{pi}, \pi) = 1$ implies

that the highest-degree coefficients vanish. Note that, in Equation (16) and Equation (17) linearity of group operations is used since the coefficients are group elements, while in Equation (18) linearity of field operations is used since the coefficients are field elements. Note that, at this point our completeness argument slightly differs from the generic argument of ProtoStar, since the homomorphism of the commitment scheme is not used for an error commitment check.

To prove knowledge soundness, we show that there exists a (D_1+1)-special-sound (public-coin) interactive protocol, and then we apply the Fiat-Shamir transform to it. The interactive protocol $(\mathcal{P}(\text{pi}, \pi, acc), \mathcal{V}(\text{pi}, \pi_x, acc_x))$ is defined as follows. $\mathcal{P}$ sends the accumulation proof

$$\text{pf} = \{\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_{D_1-1}^{(1)}, \mathbf{e}_1^{(2)}, \ldots, \mathbf{e}_{D_2-1}^{(2)}, \mathbf{e}_1^{(3)}\}$$

as computed by $\mathcal{P}_{\text{fold}}$ to $\mathcal{V}$. $\mathcal{V}$ replies with a random challenge α and then $\mathcal{P}$ responds with acc' as computed by $\mathcal{P}_{\text{fold}}$. $\mathcal{V}$ accepts if $\mathcal{D}(\text{acc}') = 1$ and $\mathcal{V}_{fold}(\text{pi}, \pi_x, \text{acc}_x, \text{acc}'_x, \text{pf}) = 1$ using the random challenge α, instead of a Fiat-Shamir challenge.

Consider $\mathcal{R}_{fold}$ defined by Relation (1) and $D_1 + 1$ accepting transcripts for $(\mathcal{P}(\text{pi}, \pi, acc), \mathcal{V}(\text{pi}, \pi_x, acc_x))$ of the form:

$$(\text{pf}, \alpha_i, acc'_i)$$

for $1 \leq i \leq D_1 + 1$ with different challenges $\alpha_1, \ldots, \alpha_{D_1+1}$, the same pf and where

$$\text{acc}'_i = \{\text{acc}'_{x,i} = \{\mathbf{V}''_i, \mathbf{c}''_i, A''_{I,i}, A''_{O,i}, S''_i, T''_{1,i}, T''_{3,i}, \ldots, T''_{6,i}, C''_{k+1,i}, y''_i, z''_i, x''_i,$$

$$E'^{(1)}_i, E'^{(2)}_i, E'^{(3)}_i, \sigma'_i\},$$

$$\text{acc}'_{w,i} = \{\tau''_{x,i}, \mu''_i, \hat{t}''_i, \mathbf{l}''_i, \mathbf{r}''_i\}\}.$$

We construct an extractor $\mathcal{E}_{fold}$ that extracts a witness for $\mathcal{R}_{fold}(\text{pi}, \pi_x, \text{acc}_x)$. Consider 2 accepting transcripts with $\alpha_1 \neq \alpha_2$, then $\mathcal{E}_{fold}$ can efficiently derive

$$\pi_w.\{\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\} = \frac{\text{acc}'_{w,1} - \text{acc}'_{w,2}}{\alpha_1 - \alpha_2},\ \text{acc}_w.\{\tau'_x, \mu', \hat{t}', \mathbf{l}', \mathbf{r}'\} = \text{acc}'_{w,1} - \alpha_1 \cdot \pi_w,$$

such that

$$\pi.C_{k+1} = Commit(\pi_w.\{\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}\})$$

and

$$acc.C'_{k+1} = Commit(acc_w.\{\tau'_x, \mu', \hat{t}', \mathbf{l}', \mathbf{r}'\}).$$

This holds by the homomorphism of the commitment scheme and because $\mathcal{D}(\text{acc}'_i)$ accepts and $\mathcal{V}_{fold}(\text{pi}, \pi_x, \text{acc}_x, \text{acc}'_{x,i}, \text{pf})$ accepts using the random challenge α_i. If for any other transcript $\text{acc}'_{w,i} \neq \alpha_i \cdot \pi_w + \text{acc}_w$ we have found a break of the commitment binding property.

Since this happens with negligible probability, this implies that the degree-D_1 polynomial

$$p_1(X) = \sum_{d=0}^{D_1} (X + \text{acc}.\sigma)^{D_1 - d} f_d^{\mathcal{V}^{\delta_{\text{new}}}}(u''(X)) - acc.E^{(1)} - \sum_{j=1}^{D_1 - 1} X^j \text{pf}.\mathbf{e}_j^{(1)} \quad (19)$$

vanishing at $D_1 + 1$ distinct challenges $\alpha_1, \ldots, \alpha_{D_1+1}$ is necessary the zero polynomial. The constant coefficient of the polynomial in Equation (19) being zero implies that

$$\text{acc}.E^{(1)} = \sum_{d=0}^{D_1} \text{acc}.\sigma^{D_1 - d} \cdot f_d^{\mathcal{V}^{\delta_{\text{new}}}}(\text{acc}.\{\mathbf{V}', \mathbf{c}', T_1', T_3', \ldots, T_6', \tau_x', \hat{t}', y', z', x'\}).$$

Moreover, the degree-D_1 coefficient of the polynomial in Equation (19) is

$$\sum_{d=0}^{D_1} f_d^{\mathcal{V}^{\delta_{\text{new}}}}(\text{pi}.\{\mathbf{V}, \mathbf{c}\}, \pi.\{T_1, T_3, \ldots, T_6, \tau_x, \hat{t}, y, z, x\}) = 0.$$

Repeating this step for $\mathcal{V}^{comm}$, we have that the degree-D_2 polynomial

$$p_2(X) = \sum_{d=0}^{D_2} (X + \text{acc}.\sigma)^{D_2 - d} f_d^{\mathcal{V}^{comm}}(u''(X)) - acc.E^{(2)} - \sum_{j=1}^{D_2 - 1} X^j \text{pf}.\mathbf{e}_j^{(2)} \quad (20)$$

vanishing at $D_2 + 1 \leq D_1 + 1$ distinct challenges $\alpha_1, \ldots, \alpha_{D_2+1}$ is necessary the zero polynomial. The constant coefficient of the polynomial in Equation (20) being zero implies that

$$\text{acc}.E^{(2)} = \sum_{d=0}^{D_2} \text{acc}.\sigma^{D_2 - d} \cdot f_d^{\mathcal{V}^{comm}}(\text{acc}.\{A_I', A_O', S', \mu', \mathbf{l}', \mathbf{r}', y', z', x'\}).$$

Moreover, the degree-D_2 coefficient of the polynomial in Equation (20) is

$$\sum_{d=0}^{D_2} f_d^{\mathcal{V}^{comm}}(\pi.\{A_I, A_O, S, \mu, \mathbf{l}, \mathbf{r}, y, z, x\}) = 0.$$

Finally, for $\mathcal{V}^{\hat{t}}$ we have that the degree-2 polynomial

$$p_3(X) = \sum_{d=0}^{2} (X + \text{acc}.\sigma)^{2 - d} f_d^{\mathcal{V}^{\hat{t}}}(u''(X)) - acc.E^{(3)} - X\text{pf}.\mathbf{e}_1^{(3)} \quad (21)$$

vanishing at $3 \leq D_1 + 1$ distinct challenges $\alpha_1, \alpha_2, \alpha_3$ is necessary the zero polynomial. The constant coefficient of the polynomial in Equation (21) being zero implies that

$$\text{acc}.E^{(3)} = \sum_{d=0}^{2} \text{acc}.\sigma^{2 - d} \cdot f_d^{\mathcal{V}^{\hat{t}}}(\text{acc}.\{\hat{t}', \mathbf{l}', \mathbf{r}'\}).$$

Moreover, the degree-2 coefficient of the polynomial in Equation (21) is

$$\sum_{d=0}^{2} f_d^{\mathcal{V}^{\hat{t}}}(\pi.\{\hat{t}, \mathbf{l}, \mathbf{r}\}) = 0.$$

The equalities of constant coefficients, together with $acc.C'_{k+1} = Commit(acc_w)$, imply that $\mathcal{D}(\text{acc})$ accepts. The equalities of highest-degree coefficients together with $\pi.C_{k+1} = Commit(\pi_w)$, imply that $\mathcal{V}_{NARK}(\text{pi}, \pi)$ accepts. Hence, the extractor $\mathcal{E}_{fold}$ computes a valid witness $(\pi_w, acc_w) \in \mathcal{R}_{fold}(\text{pi}, \pi_x, acc_x)$ and $(\mathcal{P}(\text{pi}, \pi, acc), \mathcal{V}(\text{pi}, \pi_x, acc_x))$ is $(D_1 + 1)$-special-sound. Note that, unlike the generic ProtoStar soundness argument, the extractor $\mathcal{E}_{fold}$ does not need to compute openings of committed error terms, and the polynomial argument directly relies on the explicit error terms carried in the accumulator and folding proof. Moreover, for the polynomials in Equation (19) and Equation (20), the Vandermonde argument is applied over $\mathbb{Z}_p$. Since the coefficients of these polynomials lie in a cyclic group $\mathbb{G}$ of prime order p, scalar multiplication by any non-zero element of $\mathbb{Z}_p$ is invertible in $\mathbb{G}$. This implies that the polynomial argument holds. Finally, by applying the Fiat-Shamir transform to $(\mathcal{P}(\text{pi}, \pi, acc), \mathcal{V}(\text{pi}, \pi_x, acc_x))$, we obtain a NARK for $\mathcal{R}_{fold}$. Consequently, the Bulletproofs∗ folding scheme $(\mathcal{P}_{fold}, \mathcal{V}_{fold}, \mathcal{D})$ is $((T + 1) \cdot \frac{D_1+1}{|\mathbb{Z}_p|} + \text{negl}(\lambda))$-knowledge-sound by Lemma 1.