Zero-Knowledge Proof Systems with Distributed Proof Generation

No Author Given

No Institute Given

Abstract. Zero knowledge is an intriguing area of research from the prospect of theory as well as application, where a party, prover P, tries to convince verifier V that a statement x(instance of an NP language) is correct using a witness w such that $M(x, w) = 1$. Let C be the circuit representation of M and $|C|$ is the size of C i.e. the no. of gates in C. The work of Ligero in this problem proposed a solution where proof size is $O(|C|^{1/2})$. They solve the problem by converting a NP language to it's corresponding circuit satisfiability problem. $w \in \mathbb{F}^{ml}$ be the extended witness which is the secret input to the prover. Prover P encodes w to a matrix U of size $m \times n$, where $m, n, l = O(\sqrt{|C|})$ and $n > l$. In Ligero's protocol first check is for to ensure that the matrix U is a correct encoding, that is done by testing interleaved. Then they check that all the outcomes of all the gates(addition and multiplication) are correct or not, and to do that they introduce two checks, linear constraint and quadratic constraint. We are using the same approach of Ligero, but our encoding is different. Instead of matrix we will encode w to a box(matrix of 3 dimensions), to keep the familiarity of notation we will consider $w \in \mathbb{F}^{pml}$ and the size of the box U is $p \times m \times n$, where $p, m, n, l = O(\sqrt[3]{|C|})$ and $n > l$. To encode w we are going to construct pm many polynomials(univariate) such that each of them has degree at most s, where s is some suitable number between l and n.

1 Introduction

t

[**VD:** I am using Alice - feminine - to refer to a prover, and maculine for verifier]

— Motivation for proving knowledge of witness, shared across different parties. —
— Our contributions —
— Prior work in brief —

1.1 Overview

Overview of our work: the notion, techniques. define the notion of distributed proof generation provide a protocol which satisfies these notions

Distributed proof generation

Informal notion of Distributed Provers Zero Knowledge Protocol (P,V)

A naïve approach to build a DPZK protocol is to take a single-prover ZK protocol and execute a multi-party computation among the provers of each message to be generated by the prover in the single-prover protocol. This satisfies the required security properties, but suffers in concrete efficiency when MPC is used over the proof generation as a black box.

Need for homomorphic commitments (Oracle access for IOP). The (non-interactive) zero-knowledge proof generation protocols based on the Interactive oracle proofs (IOP) or the Interactive PCP (IPCP) paradigms make use of *oracles*. The prover establishes an oracle based on her witness. Once established, the IOP/IPCP protocols ensure that a prover will not modify the oracle further in the protocol. The verifier queries the oracle later in the protocol to verify some the claims made by the prover. When IOP/IPCP based proof systems are converted into the non-interactive setting, the establishment of the oracle is done by producing a commitment to the oracle entries.

We will now run through a mental experiment on achieving distributed proof generation following the blueprint of the single-prover proof generation algorithm. We desire to not have the proof length depend on the number of parties. As we would discuss in depth later, we would also avoid running an MPC protocol on the proof generation circuit(s) of some zero-knowledge protocol. This would trivially, in theory, satisfy our goals, but it would be very inefficient in practice. The way we satisfy these goals for the establishment of the oracle is by requiring the oracle entries and their commitments satisfy homomorphic properties.
[**VD:** Very weak argument above. Make it more compelling.]

Technical summary

We will now provide a high-level summary of our work and along the way mention our core ideas.

Our contributions are two-fold: first in defining and constructing a zero-knowledge protocol with distributed proof generation and the second in proposing the first protocol with (expected) sub-linear verifier complexity.

We design a framework for constructing DPZK protocols starting from a single-prover ZK protocol. Let the parties $P_1, \ldots, P_p$ hold their respective witnesses $w_1, \ldots, w_p$ to the relation $R(x, (w_1, \ldots, w_p))$. Starting with a (public coin honest verifier) single-prover ZK protocol, there are two steps involved in our proof generation.

- The first step involves performing a secret sharing based MPC protocol on R between the provers. As a result, each prover obtains her share of the extended witness, which includes all the wires (correspondingly variables) of a circuit (corr. R1CS) representation of R.
- Next, for each message from the prover to the verifier, the parties perform the following two steps:
  - The parties run an MPC protocol to obtain their share of the message. This step has a major implication on the efficiency of distributed proof generation. Depending on the single-prover protocol, some messages already posses homomorphism properties. Hence, the output of a local run of the single-prover algorithm for this round on a prover's shares of extended witness is her share of the message in the distributed proof generation. When not homomorphic, we design custom efficient protocols to obtain the shares of the message.
  - One of the parties or an external entity untrusted with the privacy of the inputs is chosen as an "aggregator". The aggregator receives the shares of the messages from all the parties and performs the steps of the single-prover protocol to be performed with the aggregated message.

Starting with the single-prover protocol of [**?**], we design custom protocols so that the only additional interaction between the parties during the second step involves MPC for $\bar{N}$ concurrent multiplications, where $\bar{N}$ is the number of wires in the *interactive* part of the circuit. Consider the check $x + y \stackrel{?}{=} z$ when given their respective hashes. If the inputs are present with three different parties, the corresponding circuit for the hash check followed by the addition and equality check involves extensive local computations for the hash before the interaction happens.
[**VD:** Provide a compelling real world use-case here]
The additional communication complexity of our DPZK protocol grows only with the size of the interactive part. To enable this, we do.... start with [**?**] — arrange the rows accordingly in the witness matrix —

proof size independent of number of parties — Oracle queries outputs should be independent of the number of parties — we do it having parties generate a homomorphic oracle starting with their shares on the extended witness — we would achieve indistinguishability between proof generated in a distributed manner and the proof generated by a single party — In this, we use homomorphic commitments — we follow from [**?**] —

Let P be the predicate that is being proved. An aggregation of multiple proofs on the same P produces a proof. Let us consider the interactive version of the protocol. A proof consists of the messages sent by the prover to the verifier and, additionally in IOPs, the replies to the oracle queries of the verifier. where each message from the prover is first identified whether it is homomorphic or not. If each prover can produce an additive share of the proof. find a way to *aggregate* the proof parts from each prover. identify the prover messages which are homomorphic.

the goals that we set for our construction

the theme that we will have - require every message from prover to verifier be homomorphic.

concrete starting point - ligero —

2 Preliminaries

2.1 Codes

Reed-Solomon Code: For positive integers n, k, finite field $\mathbb{F}$, and a vector $\eta = (\eta_1, \cdots, \eta_n) \in \mathbb{F}_n$ of distinct field elements, the code $RS_{\mathbb{F},n,k,\eta}$ is the $[n, k, n-k+1]$ linear code over $\mathbb{F}$ that consists of all n-tuples $(p(\eta_1), ..., p(\eta_n))$ where $p(\cdot)$ is a polynomial of degree $< k$ over $\mathbb{F}$.
[**VD:** 1. Have spaces between paragraphs even in Latex. The PDF generated by Latex is usually beautiful and readable. The tex file can't be beautiful but should atleast be as readable as the PDF.
2. I have already told you many time, and I am not going to leave till you use it :) Use macro for any notation that you use more than two times throughout the paper. (You will realize the use of it the day we decide change some notation in the middle of writing a paper.. But it is also a good practice to do it in general).
3.Use ldots for ...
4. You could use subsection* or subsubsection* instead of having a paragraph and using a textbf inside. (try both, what looks better depends on the cls file you use).]

Interleaved Code: Let $L \subset \mathbb{F}_n$ be an $[n, k, d]$ linear code over $\mathbb{F}$. We let L^m denote the $[n, mk, d]$ (interleaved) code over $\mathbb{F}^m$ whose code words are all $m \times n$ matrices U such that every row U_i of U satisfies $U_i \in L$. For $U \in L^m$ and $j \in [n]$, we denote by $U[j]$ the j^{th} symbol (column) of U.
[**VD:** Have ”th” in jth outside math mode.]

2.2 Interactive Oracle Proofs

The Interactive Oracle Proofs is the notion which combine both Interactive Proofs and Probabilistically Checkable Proofs, and also generalize the notion of the Interactive PCPs.

A k-round public-coin IOP has k rounds of interaction. In the i^{th} round of interaction, the verifier sends a uniformly random message m_i to the prover; then the prover replies with a message π_i to the verifier. After k rounds of interaction, the verifier makes some queries to the oracles it received and either accepts or rejects.
[**VD:** Have at least the main definitions in the Definition environment.]

An IOP system for a relation $\mathcal{R}$ with round complexity k and soundness error ϵ is a pair (P, V), where P, V are probabilistic algorithms, that satisfies the following properties:

Completeness: For every instance-witness pair (x, w) in the relation $\mathcal{R}$, $(P(x, w), V(x))$ is a $k(n)$-round interactive oracle protocol with accepting probability 1.

Soundness: For every instance $x \notin \mathcal{L}(\mathcal{R})$ and unbounded malicious prover P^*, $(P^*, V(x))$ is a $k(n)$-round interactive oracle protocol with accepting probability at most $\epsilon(n)$.

2.3 Zero-Knowledge

Interactive Argument Systems: A pair of PPT(Probabilistic Polynomial Time) interactive machines $< P, V >$ is called an interactive proof system for a language $\mathcal{L}$ if there exists a negligible function $negl(\cdot)$ such that the following two conditions hold:
[**VD:** 1. have space before a (or any other bracket.
2. Use langle and rangle instead of ¡ and ¿ when using it as brackets. 3. You can have negl in mathsf, it would look better (have a macro for this too!)]

(1) *Completeness:* For every $x \in \mathcal{L}$ there exists a string w such that for every $z \in \{0,1\}^*$, $Pr[< P(x, w), V(x, z) >= 1] \geq 1 - negl(|x|)$.
(2) *Soundness:* For every $x \notin \mathcal{L}$, every interactive PPT machine P^*, and every $w, z \in \{0,1\}^*$ $Pr[< P^*(x, w), V(x, z) >= 1] \leq negl(|x|)$

Zero Knowledge: Let$< P, V >$ be an interactive proof system for some language $\mathcal{L}$. We say that $< P, V >$ is computational zero-knowledge with respect to an auxiliary input if for every PPT interactive machine V^* there exists a PPT algorithm S, running in time polynomial in the length of its first input, such that $\{< P(x, w), V^*(x, z) > \}_{x\in\mathcal{L}, w\in\mathcal{R}_x, z\in\{0,1\}^*} \approx_c \{< S(x, z) >\}_{x\in\mathcal{L}, z\in\{0,1\}^*}$

2.4 Commitment schemes

Commitemnts: A non-interactive commitment scheme consists of a pair of probabilistic polynomial time algorithms $(Setup, Com)$. The setup algorithm $pp \leftarrow Setup(1^\lambda)$ generates public parameters pp for the scheme, for security parameter λ. The commitment algorithm Com_{pp} defines a function $M_{pp} \times R_{pp} \rightarrow C_{pp}$ for message space M_{pp}, randomness space R_{pp} and commitment space C_{pp} determined by pp. For a message $x \in M_{pp}$, the algorithm draws $\delta \in_R R_{pp}$ uniformly at random, and computes commitment $\mathsf{com} = Com_{pp}(x; \delta)$.
For ease of notation we write $Com = Com_{pp}$.
[**VD:** All algorithm names in mathsf (macro for each). Eg. in the above paragraph, Setup, Com, ...]

Homomorphic Commitment: A homomorphic commitment scheme is a non-interactive commitment scheme such that M_{pp}, R_{pp} and C_{pp} are all abelian groups, and for all $x_1, x_2 \in M_{pp}, \delta_1, \delta_2 \in R_{pp}$, we have $Com(x_1; \delta_1) + Com(x_2; \delta_2) = Com(x_1 + x_2; \delta_1 + \delta_2)$
[**VD:** Hiding and binding are the core properties of a commitment scheme, i.e., a definition of a commitment scheme includes the properties of hiding and binding. So bring them first. And just mention “Hiding” and “Binding”.]

Hiding Commitment: A commitment scheme is said to be hiding if for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\mu(\lambda)$ such that

$$|Pr[b = b' | pp \leftarrow Setup(1^\lambda); (x_0, x_1) \in M_{pp}^2 \leftarrow \mathcal{A}(pp), b \in_R \{0, 1\}, \delta \in_R R_{pp}, \mathsf{com} = Com(x_b; \delta), b' \leftarrow \mathcal{A}(pp, com)] - \frac{1}{2}| \leq \mu(\lambda)$$

Binding Commitment: A commitment scheme is said to be binding if for all PPT adversaries $\mathcal{A}$ there exists a negligible function μ such that

$$Pr[Com(x_0; \delta_0) = Com(x_1, \delta_1) \wedge x_0 \neq x_1 | pp \leftarrow Setup(1^\lambda) x_0, x_1, \delta_0, \delta_1 \leftarrow \mathcal{A}(pp)] \leq \mu(\lambda)$$

[**VD:** Have a subsection* with Pedersen commitment, and have the vector commitment also in the same part.]

Pedersen Commitment: $M_{pp}, R_{pp} = \mathbb{Z}_p, C_{pp} = \mathbb{G}$ of order p.
$Setup : g, h \in_R \mathbb{G}$
$Com(x, \delta) = (g^x h^\delta)$

Pedersen Vector Commitment: $M_{pp} = \mathbb{Z}_p^n, R_{pp} = \mathbb{Z}_p, C_{pp} = \mathbb{G}$ with G of order p.
$Setup : \mathbf{g} = (g_1, \cdots, g_n), h \in_R \mathbb{G}$
$Com(\mathbf{x} = (x_1, \cdots, x_n); \delta) = h^r \mathbf{g}^{\mathbf{x}} = h^r \prod_i g_i^{x_i} \in \mathbb{G}$

3 Distributed Prover Zero Knowledge

3.1 Definition

Discussion Discussion on the definition.

3.2 Construction step 1: The introduction of homomorphic commitments

– some technical description of the techniques, and how "commitment oracle" helps in realizing DPZK

Note If we did not use homomorphic commitment we get +poly(secp).N protocol.

4 DPZK with reduced proof size

An brief recall on how our paradigm enables any improvement in single prover ZK protocol directly translate to the DP version.

4.1 Construction step 2: Skewing the witness matrix

ZK with $O(n^{1/c})$ proof size and $O(n^{1-1/c})$ pub key operations for verifier. — Using skewed matrix and inner product arguments.

5 DPZKQuickVerify

$O(n^{1/c})$ proof with efficient verifier.

5.1 Non-interactive

IOP to NIROP

6 Instantiations

(For c=3,4)

- Instantiate c=3,4 with square-root inner product argument.
- Verifier even more efficient than (4).
- Low round complexity

7 Implementation and Evaluation

Prior content from below

8 Construction

Encoding: Let, $x \in \mathbb{F}^{pml}$ be the secret. We will encode x to $U \in \mathbb{F}^{pmn}$ in the following way.
Let

Construct polynomials $p_{ij}(\cdot)$ of degree $< s$ such that $p_{ij}(\zeta_k) = x_{ijk} \ \ \forall i \in [p], j \in [m], k \in [l]$.
Define: U

$$U[i] = \begin{bmatrix} p_{i1}(\eta_1) & p_{i1}(\eta_2) & \cdots & p_{i1}(\eta_n) \\ p_{i2}(\eta_1) & p_{i2}(\eta_2) & \cdots & p_{i2}(\eta_n) \\ \vdots & & & \\ p_{im}(\eta_1) & p_{im}(\eta_2) & \cdots & p_{im}(\eta_n) \end{bmatrix}$$

and $U = [U[1], U[2], \cdots, U[p]]$.

So, U is a box with p slices and each slice has m rows and n columns.

Commitment: Let *com* be an homomorphic commitment scheme which commits a vector. Define:

$$com(U) = [c_1, c_2, \cdots, c_p]^T = \begin{bmatrix} c[1,1] \; c[1,2] \cdots \; c[1,n] \\ c[2,1] \; c[2,2] \cdots \; c[2,n] \\ \vdots \\ c[p,1] \; c[p,2] \cdots \; c[p,n] \end{bmatrix} = C$$

where $c[i,k] = com([p_{i1}(\eta_k), p_{i2}(\eta_k), \cdots, p_{im}(\eta_k)]^T)$ and let c is the Merkle tree root of C where leaves are columns of C.

Protocol for Testing Interleaved: P and V does the following:

1. P sends c, the Merkle root of the commitment C, to V.
2. $V \to P$: $r \in_R \mathbb{F}^p$
3. P: $\widetilde{U} = \sum \lim_{i\in[p]} r_i U[i]$, $\tilde{c} = \mathsf{com}(\widetilde{U})$
4. $P \to V$: $\tilde{c}$
5. $V \to P$: $\gamma \in_R \mathbb{F}^m$
6. P: $w = \gamma^T \widetilde{U}$
7. $P \to V$: w
8. $V \to P$: $Q \subset_R [n] : |Q| = t$
9. $P \to V$: $\widetilde{U}[k]$ *with the randomness* δ_k *to commit* $\widetilde{U}[k] : k \in Q$
10. $P \to V$: $C[k]$ *with the corresponding Merkle paths to the root* $: k \in Q$

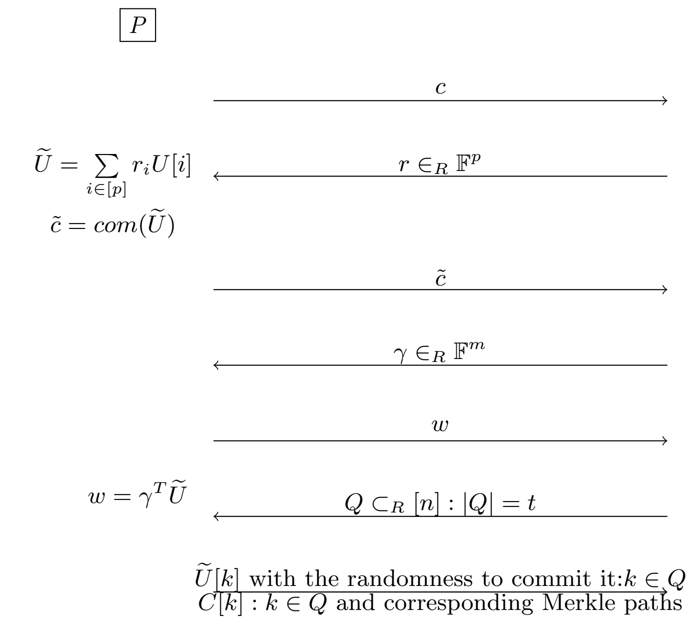
P
V
c
$\widetilde{U} = \sum_{i\in[p]} r_i U[i]$
$r \in_R \mathbb{F}^p$
$\tilde{c} = com(\widetilde{U})$
$\tilde{c}$
$\gamma \in_R \mathbb{F}^m$
w
$w = \gamma^T \widetilde{U}$
$Q \subset_R [n] : |Q| = t$
$\widetilde{U}[k]$ with the randomness to commit it:$k \in Q$
$C[k] : k \in Q$ and corresponding Merkle paths

Fig. 1. Protocol for Testing Interleaved

P and V follows the protocol in Figure 1 and then V checks the following:

(a) Validation of $C[k]$ with respect to c.

(b) $\tilde{c}_k = \sum_{i \in [p]} r_i c_{ik}\ \forall k \in Q$

(c) $w \in L$, L is the set of codewords.

(d) $\sum_{j \in [m]} \gamma_j \widetilde{U}[j,k] = w_k\ \forall k \in Q'$

(e) $open(\tilde{c}_k) = \widetilde{U}[k]\ \forall k \in Q'$

Completeness: It is easy to see that if P has correct witness, for any choice of r, Q, γ, Q' of V, P can response for which V accepts.

Soundness: The following lemma ensures the soundness:

Lemma 1. *Suppose $d(U^*, L^{mp}) > e$, where e is some positive integer such that $e < \frac{d}{3}$(what is the bound of e), U^* is not a correct encoding then any adversarial prover strategy succeeds with probability at most $(1-\frac{e}{2n})^\kappa + \frac{d}{|\mathbb{F}|} + neg(\kappa)$.*

Proof. Let $\overline{U} = \sum_{i \in [p]} r_i U^*[i]$.

Then with probability $\geq 1 - \frac{d}{|\mathbb{F}|}$, $\overline{U}$ satisfies $d(\overline{U}, L^m) > e$.

Let $\overline{c}$ be the commitment for $\overline{U}$, that implies $\overline{c} = r^T\overline{C}$, by the homomorphic property of the commitment scheme, where $\overline{C}$ is the commitment of $\overline{U}$.

Define: Let c, c' be two vectors, the distance between these two vectors is denoted by $d(c, c')$ and defined by the number of positions where c and c' differs.

case 1: $d(\overline{c}, \tilde{c}) \geq \frac{e}{2}$ where $\tilde{c} = r^T C$, where C is the commitment of U, nearest element of U^* in L^{mp}.

Prover's success probability $\leq \frac{\binom{n-e/2}{t}}{\binom{n}{t}} \approx (1-\frac{e}{2n})^t \approx (1-\frac{e}{2n})^k$ since $t = O(k)$

case 2: $d(\overline{c}, \tilde{c}) < \frac{e}{2}$.

Let, $\overline{\Delta} = \Delta(\overline{U}, L^m)$, positions where $\overline{U}$ is different from the nearest correct code slice.

and let $\partial = \Delta(\overline{c}, \tilde{c})$

We have, $\overline{\Delta} > e$ and $\partial < \frac{e}{2}$, $\Rightarrow |\overline{\Delta} \setminus \partial| \geq \frac{e}{2}$.

Therefore, for $j \in \overline{\Delta} \setminus \partial$, $\overline{c}_j = \tilde{c}_j \Rightarrow \overline{U}[j] = \widetilde{U}[j]$

Therefore probability of prover's success in the check (d) is $(1-\frac{e}{2n})^t \approx (1-\frac{e}{2n})^k$ since $t = O(k)$

So the soundness error is $(1-\frac{e}{2n})^\kappa + \frac{d}{|\mathbb{F}|} + neg(\kappa)$.

Communication Complexity:

– Proof Size: constant size Merkle root, commitment vector with n components, with each component is of constant size, say λ, together $(n+1)\lambda$, t many $C[j]$ and Merkle paths, size is $t(p.\lambda + \log n)$, n sized vector w, t many $\widetilde{U}[j]$ and randomness of total size $t(n+\alpha)$, where size of the randomness is α.
  Therefore size of the proof is $\lambda(n+1) + t(p\lambda + \log n) + n + t(n+\alpha) = O(\sqrt[3]{|C|})$
– Verifier's Complexity: $O(tm(M+E))$, where M is the cost of multiplication and E is the complexity of exponentiation. So, in this construction $O(\sqrt[3]{|C|})$ many cryptographic(elliptic curve) operations are required.
– Prover's complexity: To construct commitment matrix $|C|$ may exponentiation, to construct $\widetilde{U}$, $|C|$ many multiplication, and to construct w, mn many multiplication.
  So total complexity of the Prover is $mnM + |C|(M+E) = O(|C|(M+E))$

Protocol for checking Linear Constraint: x such that $Ax = b$

P and V does the following:

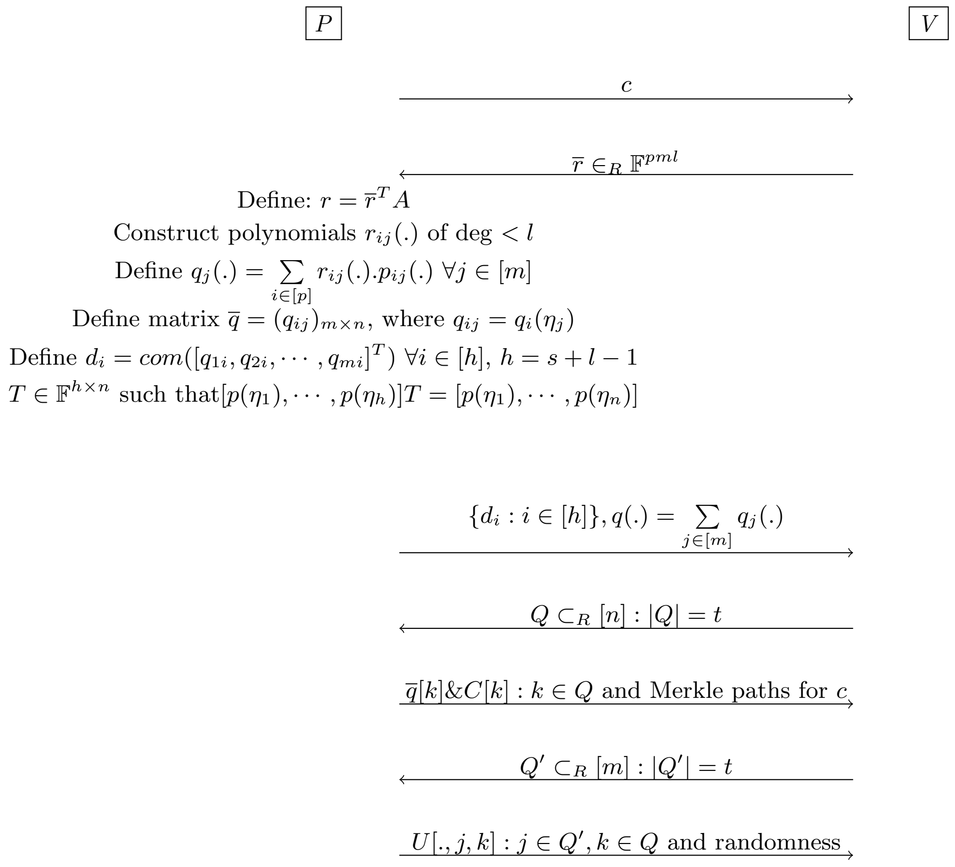
P
V
c
$\overline{r} \in_R \mathbb{F}^{pml}$
Define: $r = \overline{r}^T A$
Construct polynomials $r_{ij}(.)$ of deg $< l$
Define $q_j(.) = \sum_{i \in [p]} r_{ij}(.).p_{ij}(.)\ \forall j \in [m]$
Define matrix $\overline{q} = (q_{ij})_{m \times n}$, where $q_{ij} = q_i(\eta_j)$
Define $d_i = com([q_{1i}, q_{2i}, \cdots, q_{mi}]^T)\ \forall i \in [h]$, $h = s + l - 1$
$T \in \mathbb{F}^{h \times n}$ such that$[p(\eta_1), \cdots, p(\eta_h)]T = [p(\eta_1), \cdots, p(\eta_n)]$
$\{d_i : i \in [h]\}, q(.) = \sum_{j \in [m]} q_j(.)$
$Q \subset_R [n] : |Q| = t$
$\overline{q}[k] \& C[k] : k \in Q$ and Merkle paths for c
$Q' \subset_R [m] : |Q'| = t$
$U[., j, k] : j \in Q', k \in Q$ and randomness

Fig. 2. Protocol for checking Linear Constraint

V checks that:

(a) Validation of $C[k]$ with respect to c.
(b) $\sum_{k \in [l]} q(\zeta_k) = \overline{r}^T b$
(c) Check $q(.)$ using $d_1, \cdots, d_h$: $\langle 1^m, \overline{q}[j'] \rangle = q(\eta_{j'}) \forall j' \in [h]$ as the inner product of the commitments should match with the commitment on the value $q(\eta_{j'})$. Using the idea of bullet proofs. Note that Verifier does not have $\overline{q}[j']$ for all $j' \in [h]$ but still this check is possible using the commitments $d_1, \cdots, d_h$.
(d) Reveal$(\sum_{j' \in [h]} T_{j'k} d_{j'}) = \overline{q}[k] \forall k \in Q$
(e) Check $U[., j, k]$ with respect to $C[k]\ \forall k \in Q, j \in Q'$: $\langle e_j, U[i, ., k] \rangle = U[i, j, k]$
(f) $\sum_{i \in [p]} r_{ij}(\eta_k) U[i, j, k] = \overline{q}[k]_j\ \forall k \in Q$ and $j \in Q'$

Completeness: If the prover encodes the correct x which satisfies $Ax = b$, then all the above checks will be passed, which ensures the completeness property.

Soundness: The following lemma ensures the soundness:

Lemma 2. *Let e be a positive integer such that $e < \frac{d}{3}$. Suppose that a malformed U^* is e-close to a codeword $U \in L^{mp}$ encoding $x \in \mathbb{F}^{pml}$ such that $Ax = b$. Then, for any malicious P strategy, U^* is rejected by V except with at most $((e + k + l)/n)^\kappa + 1/|\mathbb{F}| + neg(\kappa)$ probability.*

Proof. Since we assume that $Ax \neq b$, therefore $Pr_r[r^T Ax = r^T b]$ is at most $\frac{1}{|\mathbb{F}|}$.
Otherwise, P must have sent a wrong q', instead of q. Since, degree of q' and q can be at most $k+l-2$, as q' is fixed after fixing the matrix $\overline{q}$, which is fixed and ensured to the verifier by providing $d_1, \cdots, d_h$. q and q' could be same in at most $k+l-2$ many indices of η. Let, $\overline{Q}$ is the set of indices where q' and q agree. and E be the set of indices where U and U^* are different. Therefore, in check (f), prover fails if $k \in Q$ such that $k \notin \overline{Q} \cup E$. Therefore P can cheat with probability at most $\frac{\binom{e+k+l-2}{t}}{\binom{n}{t}} \approx (\frac{e+k+l}{n})^t \approx (\frac{e+k+l}{n})^\kappa$.
Therefore soundness error is $(\frac{e+k+l}{n})^\kappa + \frac{1}{|\mathbb{F}|} + neg(\kappa)$, ($neg(\kappa)$ is due to the soundness probability of the commitment scheme).

Zero Knowledge:
Communication Complexity:

– Proof Size: First P sends Merkle root, which is of constant size, say λ, then sends $(k+l-1)$ many commitments of constant size and $(k+l-1)$ many coefficients of q, therefore, $(\lambda+1) \times (k+l-1)$. $t \times m$ to send t columns of $\overline{q}$ and $t \times p\lambda$ for t columns of C, and for Merkle paths $t \times \log(pn)$. Finally t^2 many vectors from U including the randomness to commit P needs to send $t^2(p+\alpha)$. So, total size of the proof is $\sqrt[3]{|C|}$.
– Verifier's Complexity: l many polynomial evaluation of degree at most $(s+l-1)$, for that number of multiplication is $O(l(s+l-1))$,
Checking commitments by IP using Bulletproofs, $(s+l-1)\sqrt{m}$ many exponentiations.
$t(s+l-1)$ many multiplications.
Checking commitments by IP using Bulletproofs, $t\sqrt{p}$ many exponentiations.
and finally tp many multiplications.
In total: $l(s+l-1) + t(s+l-1) + tp$ multiplications and $(s+l-1)\sqrt{m} + t\sqrt{p}$ many exponentiations
– Prover's Complexity:

Protocol for checking Quadratic Constraint: x, y, z suc that $x \odot y + a \odot z = b$
P and V does the following:
V checks that:

(a) Validation of $C^\delta[k]$ with respect to c^δ, where $\delta \in \{x, y, z\}$.
(b) $\forall c \in [l], q(\zeta_c) = 0$
(c) Check $q(.)$ using $d_1, \cdots, d_h$: $<\gamma, \overline{q}[j']>= q(\eta_{j'}) \forall j' \in [h]$ as the inner product of the commitments should match with the commitment on the value $q(\eta_{j'})$. Using the idea of bullet proofs. Note that Verifier does not have $\overline{q}[j']$ for all $j' \in [h]$ but still this check is possible using the commitments $d_1, \cdots, d_h$.
(d) Reveal$(\sum_{j' \in [h]} T_{j'k} d_{j'}) = \overline{q}[k] \forall k \in Q$
(e) Check $U^\delta[., j, k]$ with respect to $C^\delta[k]$ $\forall k \in Q, j \in Q'$: $< e_j, U^\delta[i, ., k] >= U^\delta[i, j, k]$, where $\delta \in \{x, y, z\}$.
(f) $\sum_{i \in [p]} r_i [U^x[i,j,k].U^y[i,j,k] + U^a[i,j,k].U^z[i,j,k] - U^b[i,j,k]] = \overline{q}[k]_j$ $\forall k \in Q$ and $j \in Q'$

Completeness: If the prover encodes the correct x, y, z which satisfy $x \odot y + a \odot z = b$, then all the above checks will be passed, which ensures the completeness property.

Soundness: The following lemma ensures the soundness:

Lemma 3.

Zero Knowledge:
Communication Complexity:

$$\boxed{P} \qquad\qquad\qquad\qquad\qquad\qquad \boxed{V}$$

$$\xrightarrow{c^x, c^y, c^z}$$

$$\xleftarrow{r \in_R \mathbb{F}^p}$$

Define: $p_{ij}(.) = p^x_{ij}(.).p^y_{ij}(.) + p^a_{ij}(.).p^z_{ij}(.) - p^b_{ij}(.)$

Define: $q_j(.) = \sum_{i \in [p]} r_i.p_{ij}(.) \ \forall j \in [m]$

Define matrix $\overline{q} = (q_{jk})_{m \times n}$, where $q_{jk} = q_j(\eta_k)$

Define $d_i = com([q_{1i}, q_{2i}, \cdots, q_{mi}]^T) \ \forall i \in [h] \ h = 2s - 1$

$T \in \mathbb{F}^{h \times n}$ such that$[p(\eta_1), \cdots, p(\eta_h)]T = [p(\eta_1), \cdots, p(\eta_n)]$

$$\xrightarrow{\{d_i : i \in [h]\}}$$

$$\xleftarrow{\gamma \in_R \mathbb{F}^m}$$

$$\xrightarrow{q(.) = \sum_{j \in [m]} \gamma_j.q_j(.)}$$

$$\xleftarrow{Q \subset_R [n] : |Q| = t}$$

$$\xrightarrow{\overline{q}[k] \& C^\delta[k] : k \in Q \text{ and Merkle paths for } c^\delta : \delta \in \{x, y, z\}}$$

$$\xleftarrow{Q' \subset_R [m] : |Q'| = t}$$

$$\xrightarrow{U^\delta[., j, k] : j \in Q', k \in Q, \delta \in \{x, y, z\} \text{ and randomness}}$$

Fig. 3. Protocol for checking Quadratic constraint

9 Multi-Prover Setting

Let, $P_1, P_2, \cdots, P_N$ provers are there. Let w be the extended witness which is additively distributed and w_ν is the share of the witness of the νth party $\Rightarrow \sum_{\nu \in [N]} w_\nu = w$

P_ν encodes w_ν in the following way:
Let $w_\nu \in \mathbb{F}^{pml}$, then we can write

$$w_\nu = \begin{bmatrix} w_\nu[111] & w_\nu[112] & \cdots & w_{nu}[11l] \\ w_\nu[121] & w_\nu[122] & \cdots & w_\nu[12l] \\ \vdots & & & \\ w_\nu[1m1] & w_\nu[1m2] & \cdots & w_\nu[1ml] \\ \vdots & & & \\ w_\nu[pm1] & w_\nu[pm2] & \cdots & w_\nu[pml] \end{bmatrix}$$

Construct 'pm' many polynomials of degree $< s$ such that $p_{ij}(\zeta_c) = w_\nu[i,j,c] \forall i \in [p] j \in [m] c \in [l]$
Define:

$$U^{w_\nu}[i] = \begin{bmatrix} p_{i1}^{w_\nu}(\eta_1) \cdots p_{i1}^{w_\nu}(\eta_n) \\ p_{i2}^{w_\nu}(\eta_1) \cdots p_{i2}^{w_\nu}(\eta_n) \\ \vdots \\ p_{im}^{w_\nu}(\eta_1) \cdots p_{im}^{w_\nu}(\eta_n) \end{bmatrix}$$

Define:

$$U^{w_\nu} = [U^{w_\nu}[1] \cdots U^{w_\nu}[p]]$$

Let $U = \sum_{\nu \in [N]} U^{w_\nu}$.
Note that U decodes to w. So let's call this U^w
Let C^{w_ν}, C^w be the commitment of U^{w_ν}, U^w respectively as mentioned in sectioin 1. Therefore by the homomorphic property of the commitment scheme $C^w = \sum_{\nu \in [N]} C^{w_\nu}$.

Protocol for Testing Interleaved: $P \in \{P_1, \cdots, P_N\}$ is arbitrarily chosen.

V checks that:

(a) Validation of $C[k]$ with respect to the Merkle root c.
(b) $\tilde{c}_k = \sum_{i \in [p]} r_i C_{ik} \; \forall k \in Q$
(c) $w \in L$, L is the set of codewords of size $\mathbb{F}^n$
(d) $\sum_{j \in [m]} \gamma_j \widetilde{U}_{jk} = w_k \; \forall k \in Q$
(e) $Open(\tilde{c}_k) = \widetilde{U}[k] \; \forall k \in Q$

Protocol for Linear constraint: Let $Ax = b$, where $A \in \mathbb{F}^{pml \times pml}$ and $b \in \mathbb{F}^{pml}$ are publicly known and x is the secret which is additively distributed among the provers, say x_ν is the share of the νth prover.
P_ν encodes x_ν to U^ν and C^ν is the commitment of U^ν.
$P \in \{P_1, \cdots, P_N\}$ is arbitrarily chosen.

V checks that:

(a) Validation of $C[k]$ with respect to the Merkle root c.
(b) $\sum_{c \in [l]} q(\zeta_c) = \overline{r}^T b$.
(c) Check $q(\cdot)$ using $d_1, \cdots, d_h$ by InnerProduct argument using Bullet Proofs: $< 1, \overline{q}[j'] >= q(\eta_{j'}) \; \forall j' \in [h]$ where, commitment of $\overline{q}[j']$ is $d_{j'}$.
(d) Reveal($\sum_{j' \in [h]} T_{j'k} d_{j'}$) $= q(\eta_k) \forall k \in [n]$, where T is a public matrix of dimension $n \times h$ such that $T([p(\eta_1), \cdots, p(\eta_h)]^T) = ([p(\eta_1), \cdots, p(\eta_n)]^T)$ for all polynomial of degree $< h$.
(e) Check $U[\cdot, j, k]$ with respect to $C[k] \; \forall (j,k) \in Q$ by InnerProduct argument using Bullet Proofs: $< e_j, U[i, \cdot, k] >= U[i,j,k]$.
(f) $\sum_{i \in [p]} r_{ij}(\eta_k) U[i,j,k] = \overline{q}[k]_j \; \forall (j,k) \in Q$.

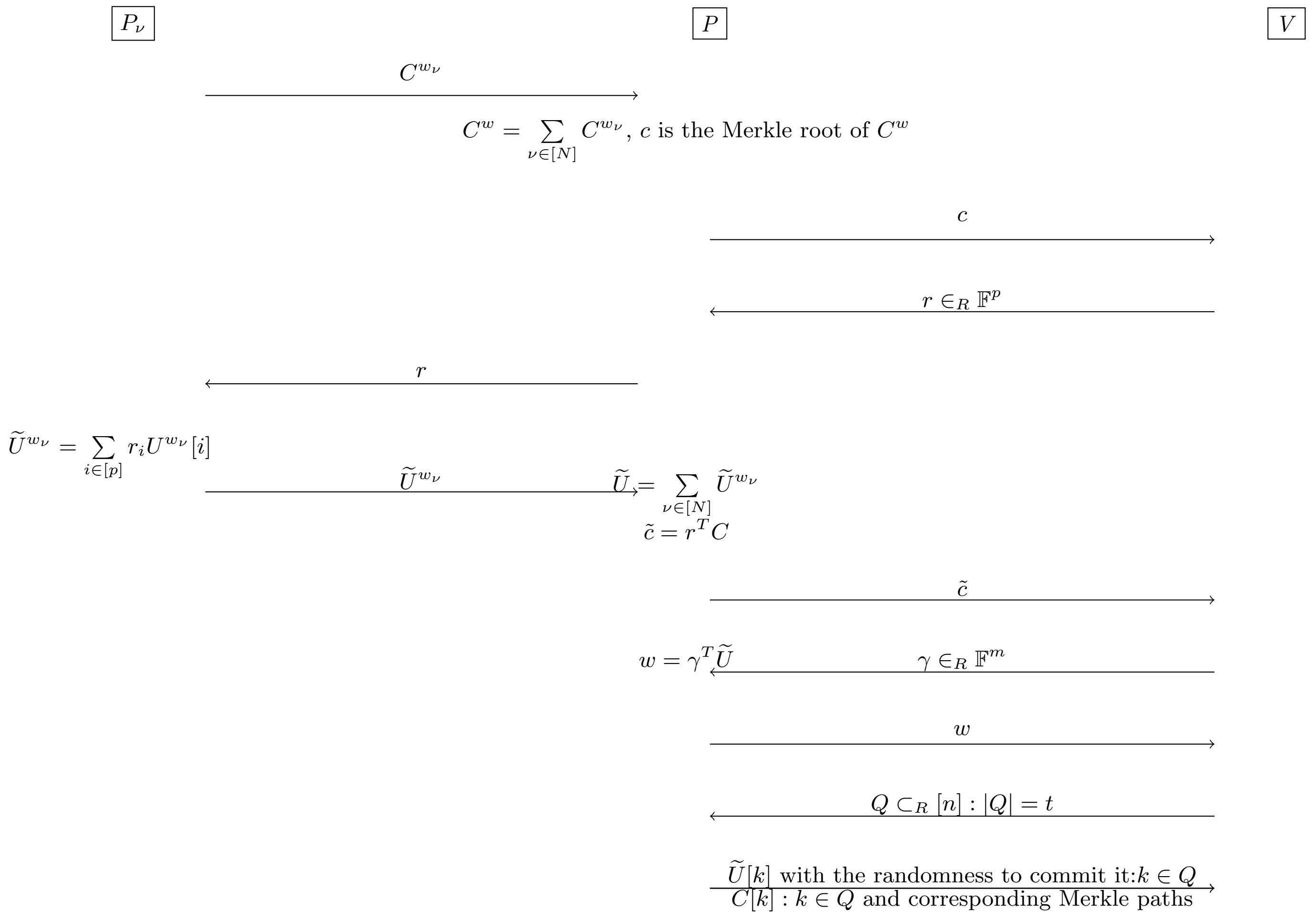
P_ν
P
V
C^{w_ν}
$C^w = \sum_{\nu \in [N]} C^{w_\nu}$, c is the Merkle root of C^w
c
$r \in_R \mathbb{F}^p$
r
$\widetilde{U}^{w_\nu} = \sum_{i \in [p]} r_i U^{w_\nu}[i]$
$\widetilde{U}^{w_\nu}$
$\widetilde{U} = \sum_{\nu \in [N]} \widetilde{U}^{w_\nu}$
$\tilde{c} = r^T C$
$\tilde{c}$
$w = \gamma^T \widetilde{U}$
$\gamma \in_R \mathbb{F}^m$
w
$Q \subset_R [n] : |Q| = t$
$\widetilde{U}[k]$ with the randomness to commit it:$k \in Q$
$C[k] : k \in Q$ and corresponding Merkle paths

Fig. 4. Protocol for Testing Interleaved

Protocol for quadratic constraint: Let $x \odot y + a \odot z = b$, where a, b are public and x, y, z are secrets. x_ν, y_ν, z_ν are the shares of P_ν. P_ν encodes x_ν, y_ν, z_ν to $U^{x_\nu}, U^{y_\nu}, U^{z_\nu}$ respectively and computes the corresponding commitments i.e. $com(U^{x_\nu}) = C^{x_\nu}, com(U^{y_\nu}) = C^{y_\nu}, com(U^{z_\nu}) = C^{z_\nu}$
P_ν constructs 'pm' many polynomials $p_{ij}^{(xy)_\nu}(\cdot)$ such that $p_{ij}^{(xy)_\nu}(\cdot) = p_{ij}^{x_\nu}(\cdot)p_{ij}^{y_\nu}(\cdot)$. To do that use FFT and choose $\{\eta_1, \eta_2, \cdots, \eta_n\}$ accordingly, in particular, nth roots of unity and set $n \geq 2s - 1$.

$P \in \{P_1, \cdots, P_N\}$ is arbitrarily chosen.

$$\boxed{P_\nu} \qquad \qquad \boxed{P} \qquad \qquad \boxed{V}$$

$$\boxed{P_\nu} \xrightarrow{C^\nu} \boxed{P}$$

$$C = \sum_{\nu\in[N]} C^\nu,\ c \text{ is the Merkle root of } C$$

$$\boxed{P} \xrightarrow{c} \boxed{V}$$

$$\boxed{P} \xleftarrow{\overline{r} \in_R \mathbb{F}^{pml}} \boxed{V}$$

$$\boxed{P_\nu} \xleftarrow{r} r = \overline{r}^T A$$

$$q_j^\nu(\cdot) = \sum_{i\in[p]} r_{ij}(\cdot) p_{ij}^\nu(\cdot)\ \forall j \in [m]$$

$$com(q_1(\eta_{j'}), \cdots, q_m(\eta_{j'})) = d_{j'} \forall j' \in [h]$$

$$q^\nu(\cdot) = \sum_{j\in[m]} q_j^\nu(\cdot) \xrightarrow{\{d_{j'}^\nu\}_{j'\in[h]}, q^\nu(\cdot)} \boxed{P}$$

$$q(\cdot) = \sum_{\nu\in[N]} q^\nu(\cdot)$$

$$d_{j'} = \sum_{\nu\in[N]} d_{j'}^\nu$$

$$\boxed{P} \xrightarrow{\{d_{j'}\}_{j'\in[h]}, q(\cdot)} \boxed{V}$$

$$\boxed{P} \xleftarrow{Q \subset_R [m] \times [n] : |Q| = t \& j_1 \neq j_2 \text{ and } k_1 \neq k_2 \forall (j_1,k_1),(j_2,k_2) \in Q} \boxed{V}$$

$$\boxed{P_\nu} \xleftarrow{Q \subset_R [m] \times [n] : |Q| = t \& j_1 \neq j_2 \text{ and } k_1 \neq k_2 \forall (j_1,k_1),(j_2,k_2) \in Q} \boxed{P}$$

$$\boxed{P_\nu} \xrightarrow{\{\{q_j^\nu(\eta_k)\}_{j\in[m]} : \forall k, \text{ for some } j, (j,k) \in Q\}} \boxed{P}$$

$$\overline{q}[k] = \{ \sum_{\nu\in[N]} q_j^\nu(\eta_k)\}_{j\in[m]}$$

$$\boxed{P} \xrightarrow{\overline{q}[k], C[k] \& U[\cdot, j, k] \forall k : (j,k) \in Q} \boxed{V}$$

Fig. 5. Protocol for Linear Constraint

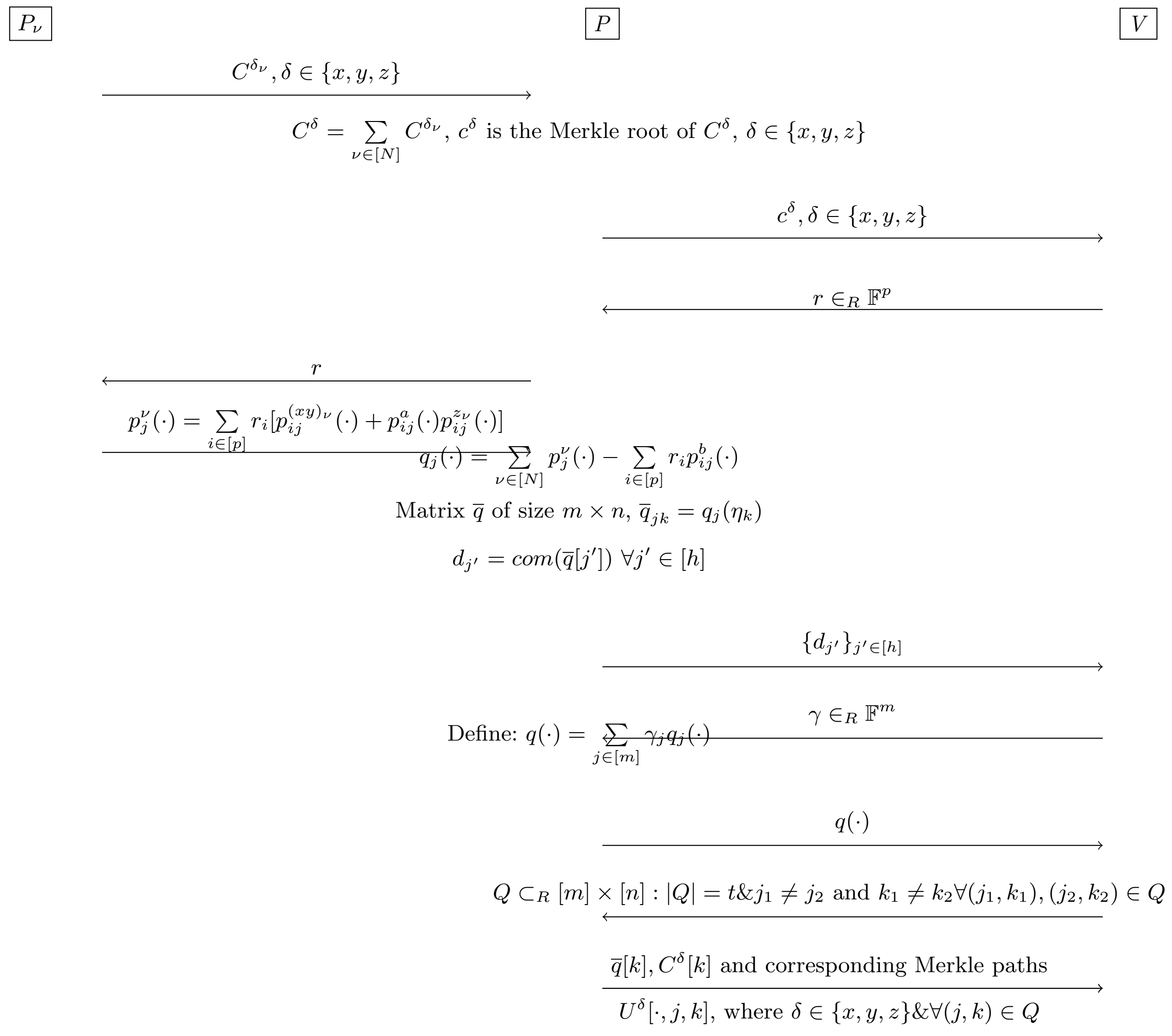
P_ν
P
V
$C^{\delta_\nu}, \delta \in \{x, y, z\}$
$C^\delta = \sum_{\nu \in [N]} C^{\delta_\nu}$, c^δ is the Merkle root of C^δ, $\delta \in \{x, y, z\}$
$c^\delta, \delta \in \{x, y, z\}$
$r \in_R \mathbb{F}^p$
r
$p_j^\nu(\cdot) = \sum_{i \in [p]} r_i[p_{ij}^{(xy)_\nu}(\cdot) + p_{ij}^a(\cdot)p_{ij}^{z_\nu}(\cdot)]$
$q_j(\cdot) = \sum_{\nu \in [N]} p_j^\nu(\cdot) - \sum_{i \in [p]} r_i p_{ij}^b(\cdot)$
Matrix $\overline{q}$ of size $m \times n$, $\overline{q}_{jk} = q_j(\eta_k)$
$d_{j'} = com(\overline{q}[j'])\ \forall j' \in [h]$
$\{d_{j'}\}_{j' \in [h]}$
$\gamma \in_R \mathbb{F}^m$
Define: $q(\cdot) = \sum_{j \in [m]} \gamma_j q_j(\cdot)$
$q(\cdot)$
$Q \subset_R [m] \times [n] : |Q| = t \& j_1 \neq j_2$ and $k_1 \neq k_2 \forall (j_1, k_1), (j_2, k_2) \in Q$
$\overline{q}[k], C^\delta[k]$ and corresponding Merkle paths
$U^\delta[\cdot, j, k]$, where $\delta \in \{x, y, z\} \& \forall (j, k) \in Q$

Fig. 6. Protocol for Quadratic Constraint

V checks that:

(a) Validation of $C^{\delta}[k]$ with respect to c^{δ}, $\delta \in \{x, y, z\}$.
(b) $q(\zeta_c) = 0 \ \forall c \in [l]$.
(c) Check using $d_1, \cdots, d_h$ using Inner Product argument by Bullet Proofs: $< \gamma, \overline{q}[j'] >= q(\eta_{j'}) \forall j' \in [h]$.
(d) Reveal($\sum_{j' \in [h]} T_{j'k} d_{j'}) = q(\eta_k) \forall k \in [n]$, where T is a public matrix of dimension $n \times h$ such that $T([p(\eta_1), \cdots, p(\eta_h)]^T) = ([p(\eta_1), \cdots, p(\eta_n)]^T)$ for all polynomial of degree $< h$.
(e) Check $U^{\delta}[\cdot, j, k]$ with respect to $C^{\delta}[k] \ \forall (j,k) \in Q$ by Inner Product argument using Bullet Proofs: $< e_j, U^{\delta}[i, \cdot, k] >= U^{\delta}[i, j, k]$.
(f) $\sum_{i \in [p]} r_i [U^x[i,j,k].U^y[i,j,k] + U^a[i,j,k].U^z[i,j,k] - U^b[i,j,k]] = \overline{q}[k]_j \ \forall (j,k) \in Q$.

10 Proof of size $|C|^{\frac{1}{\alpha}}$

Here consider our extended witness $w \in \mathbb{F}^{pm_1 m_2 l}$. Encode it to U^w in the same way as above, where size of U is $p \times m_1 m_2 \times n$. Let, $m = m_1 m_2$

Protocol for Testing Interleaved: P and V does the following:

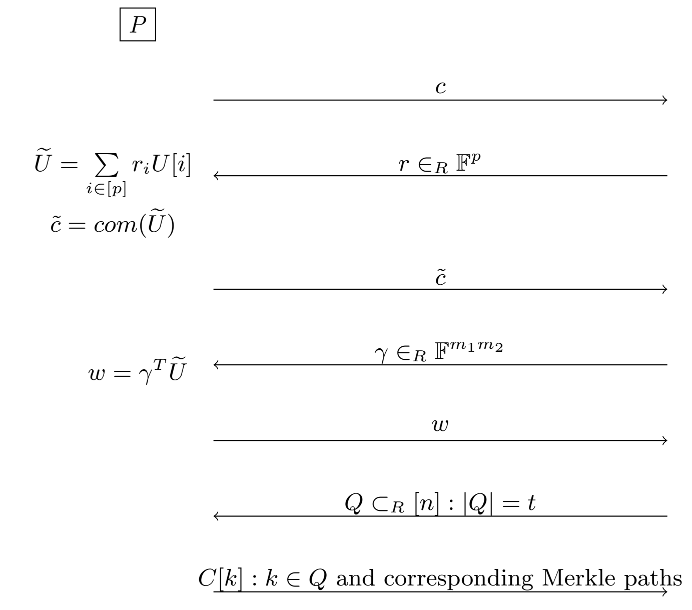
P
V
c
$\widetilde{U} = \sum_{i \in [p]} r_i U[i]$
$r \in_R \mathbb{F}^p$
$\tilde{c} = com(\widetilde{U})$
$\tilde{c}$
$w = \gamma^T \widetilde{U}$
$\gamma \in_R \mathbb{F}^{m_1 m_2}$
w
$Q \subset_R [n] : |Q| = t$
$C[k] : k \in Q$ and corresponding Merkle paths

Fig. 7. Protocol for Testing Interleaved

P and V follows the protocol in Figure 1 and then V checks the following:

(a) Validation of $C[k]$ with respect to c.
(b) $\tilde{c}_k = \sum_{i \in [p]} r_i c_{ik} \ \forall k \in Q$
(c) $w \in L$, L is the set of codewords.
(d) $\sum_{j \in [m]} \gamma_j \widetilde{U}[j,k] = w_k \ \forall k \in Q'$ Checking $\widetilde{U}$ with respect to it's commitment and γ using inner product argument i.e. $< \gamma, \widetilde{U}[k] >= w_k \ \forall k \in [n]$

Proof Size:

- constant size Merkle root
- n sized vector $\tilde{c}$
- n sized vector w
- t many p sized vector $C[k] : k \in Q$
- proof for inner product argument $O(log(m)$

So total size of the proof is $O(|C|^{\frac{1}{4}})$.

Verifier's Complexity:

- Checking Merkle root
- p multiplications: pM
- Multiplication by the Parity check matrix: $n^2 M$
- Inner Product check :$m.E$

So total $mE + pM + n^2 M$

Prover's Complexity:

- Computing C, $|C|$ many exponentiation: $|C|.E$
- Computing Merkle root
- To compute $\widetilde{U}$, $|C|.M$ and to compute $\tilde{c}$, $p.n.M$
- To compute w, $m.n.M$

So total $|C|.E + |C|.M + p.n.M + m.n.M + h$, where h is the complexit to compute the Merkle root.

Protocol for checking Linear Constraint: x such that $Ax = b$
P and V does the following:

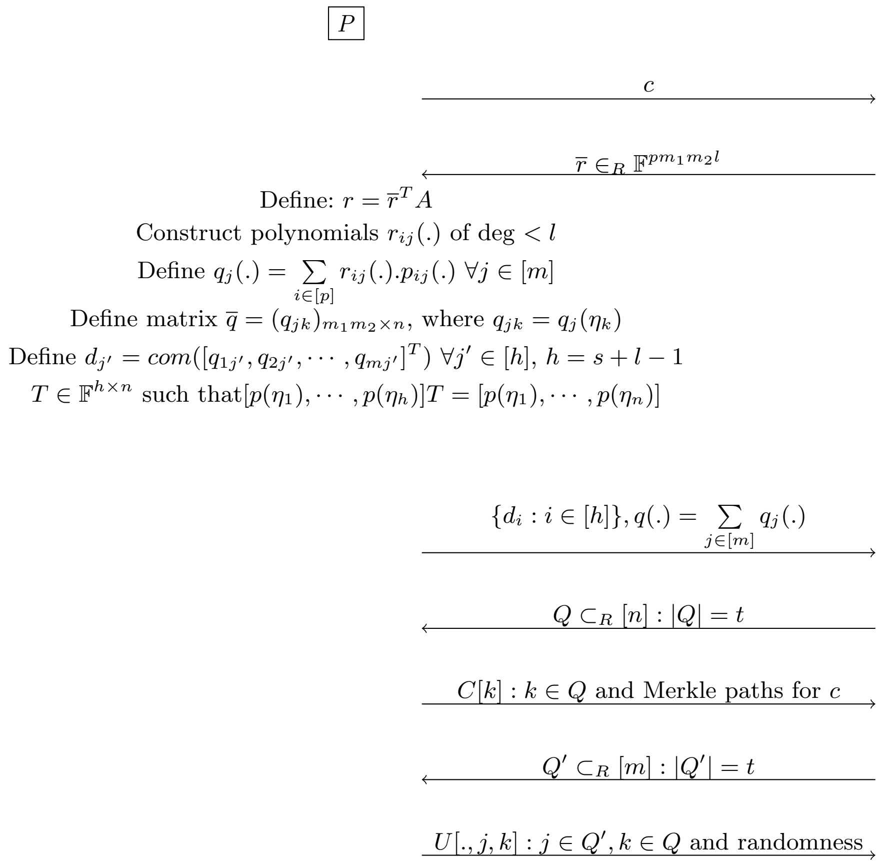
$\boxed{P}$
$\boxed{V}$
c
$\overline{r} \in_R \mathbb{F}^{pm_1m_2l}$
Define: $r = \overline{r}^T A$
Construct polynomials $r_{ij}(.)$ of deg $< l$
Define $q_j(.) = \sum_{i\in[p]} r_{ij}(.).p_{ij}(.)\ \forall j \in [m]$
Define matrix $\overline{q} = (q_{jk})_{m_1m_2\times n}$, where $q_{jk} = q_j(\eta_k)$
Define $d_{j'} = com([q_{1j'}, q_{2j'}, \cdots, q_{mj'}]^T)\ \forall j' \in [h]$, $h = s + l - 1$
$T \in \mathbb{F}^{h\times n}$ such that$[p(\eta_1), \cdots, p(\eta_h)]T = [p(\eta_1), \cdots, p(\eta_n)]$
$\{d_i : i \in [h]\}, q(.) = \sum_{j\in[m]} q_j(.)$
$Q \subset_R [n] : |Q| = t$
$C[k] : k \in Q$ and Merkle paths for c
$Q' \subset_R [m] : |Q'| = t$
$U[., j, k] : j \in Q', k \in Q$ and randomness

Fig. 8. Protocol for checking Linear Constraint

V checks that:

(a) Validation of $C[k]$ with respect to c.
(b) $\sum_{k\in[l]} q(\zeta_k) = \overline{r}^T b$
(c) Check $q(.)$ using $d_1, \cdots, d_h$: $< 1, \overline{q}[j'] >= q(\eta_{j'}) \forall j' \in [h]$ as the inner product of the commitments should match with the commitment on the value $q(\eta_{j'})$. Using the idea of bullet proofs. Note that Verifier does not have $\overline{q}[j']$ for all $j' \in [h]$ but still this check is possible using the commitments $d_1, \cdots, d_h$.
(d) Check $U[., j, k]$ with respect to $C[k]\ \forall k \in Q, j \in Q'$: $< e_j, U[i, ., k] >= U[i, j, k]$
(e) V computes the value $\sum_{j'\in[h]} T_{j'k} d_{j'}$ which is the commitment of the vector $\overline{q}[k]$ and also computes $\sum_{i\in[p]} r_{ij}(\eta_k) U[i, j, k]$. Now use Inner Product argument i.e. $< e_j, \overline{q}[k] >= \sum_{i\in[p]} r_{ij}(\eta_k) U[i, j, k]\ \forall j \in Q', k \in Q$.

Proof Size:

– Constant size Merkle root.
– h many constants for $d_1, \cdots, d_h$ and h coefficients for the polynomial, $O(h)$.
– t many $C[k]$ and corresponding Merkle paths, total size: $t.p + t \log(pn)$
– t many $U[\cdot, j, k]$ and randomness: $O(t.p)$.
– Proof for Inner Product argument $O(\log(m))$

In total $O(|C|^{\frac{1}{4}})$

Verifier's Complexity:

- Checking Merkle roots.
- l polynomial evaluations $(s+l-1).l.M$
- Inner Product check by the verifier: $O(m.E)$
- $(s+l-1).M + p.M + |C|.M + \log m.E$

So total $(s+l-1).l.M + m.E + (s+l-1).M + p.M + |C|.M + \log m.E$

Prover's Complexity: To compute C matrix, Prover needs to do $|C|$ many exponentiation and that is dominating term in his computation.

Protocol for checking Quadratic Constraint: x, y, z suc that $x \odot y + a \odot z = b$
P and V does the following:

$$\boxed{P} \qquad\qquad\qquad\qquad \boxed{V}$$

$$\xrightarrow{\quad c^x, c^y, c^z \quad}$$

$$\xleftarrow{\quad r \in_R \mathbb{F}^p \quad}$$

Define: $p_{ij}(.) = p^x_{ij}(.).p^y_{ij}(.) + p^a_{ij}(.).p^z_{ij}(.) - p^b_{ij}(.)$

Define: $q_j(.) = \sum_{i \in [p]} r_i.p_{ij}(.) \; \forall j \in [m]$

Define matrix $\overline{q} = (q_{jk})_{m_1 m_2 \times n}$, where $q_{jk} = q_j(\eta_k)$

Define $d_{j'} = com([q_{1j'}, q_{2j'}, \cdots, q_{mj'}]^T) \; \forall j' \in [h] \; h = 2s-1$

$T \in \mathbb{F}^{h \times n}$ such that$[p(\eta_1), \cdots, p(\eta_h)]T = [p(\eta_1), \cdots, p(\eta_n)]$

$$\xrightarrow{\quad \{d_i : i \in [h]\} \quad}$$

$$\xleftarrow{\quad \gamma \in_R \mathbb{F}^m \quad}$$

$$\xrightarrow{\quad q(.) = \sum_{j \in [m]} \gamma_j.q_j(.) \quad}$$

$$\xleftarrow{\quad Q \subset_R [n] : |Q| = t \quad}$$

$$\xrightarrow{\quad C^\delta[k] : k \in Q \text{ and Merkle paths for } c^\delta : \delta \in \{x, y, z\} \quad}$$

$$\xleftarrow{\quad Q' \subset_R [m] : |Q'| = t \quad}$$

$$\xrightarrow{\quad U^\delta[., j, k] : j \in Q', k \in Q, \delta \in \{x, y, z\} \text{ and randomness} \quad}$$

Fig. 9. Protocol for checking Quadratic constraint

V checks that:

(a) Validation of $C^\delta[k]$ with respect to c^δ, where $\delta \in \{x, y, z\}$.
(b) $\forall k \in [l], q(\zeta_k) = 0$
(c) Check $q(.)$ using $d_1, \cdots, d_h$: $< \gamma, \overline{q}[j'] >= q(\eta_{j'}) \forall j' \in [h]$ as the inner product of the commitments should match with the commitment on the value $q(\eta_{j'})$. Using the idea of bullet proofs. Note that Verifier does not have $\overline{q}[j']$ for all $j' \in [h]$ but still this check is possible using the commitments $d_1, \cdots, d_h$.

(d) Check $U^{\delta}[.,j,k]$ with respect to $C^{\delta}[k]$ $\forall k \in Q, j \in Q'$: $< e_j, U^{\delta}[i,.,k] >= U^{\delta}[i,j,k]$, where $\delta \in \{x,y,z\}$.
(e) V computes the value $\sum\limits_{j' \in [h]} T_{j'k} d_{j'}$, which is the commitment of the vector$\overline{q}[k]$ and also computes $\sum\limits_{i \in [p]} r_i [U^x[i,j,k].U^y[i,j,k] + U^a[i,j,k].U^z[i,j,k] - U^b[i,j,k]] = R[i,j,k]$. Now use Inner Product argument i.e. $< e_j, \overline{q}[k] >= R[i,j,k]$ $\forall j \in Q', k \in Q$.

Proof Size:

- Constant size Merkle root c.
- $d_1, \cdots, d_h$: $O(h)$.
- To send the polynomial $q(\cdot)$, $2s-1$ many coefficients are needed to be send.
- t many $C[k]$, $t.p$
- t many $U[\cdot,j,k]$, $t.p$
- Proof for Inner product argument: $\log m$

In total $O(h + 2s - 1 + t.p + \log(m) = O(|C|^{\frac{1}{4}})$

Verifier's Complexity:

- Checking Merkle roots.
- l polynomial evaluations: $(2s-1).l.M$
- Checking using Inner Product arguments $O(m.E)$
- $(2s-1).M + t.p.M$

So total complexity $O(|C|^{\frac{1}{2}}.E + |C|^{\frac{1}{2}}.M)$

Prover's Complexity: To compute C matrix, Prover needs to do $|C|$ many exponentiation and that is dominating term in his computation.

10.1 Zero-Knowledge:

We have three checks:

- Interleaved Testing
- Linear Check
- Quadratic Check

Note that during the interleaved testing, w is sent to the verifier, where w is the linear combination of all the rows of the encoding matrix, which verifier cannot compute on his own. So, it is required to blind the vector w by adding an additional vector which is a codeword.

For Linear check, verifier is receiving the polynomial $q(\cdot)$, and can evaluates in many points which he should not learn on his own, verifier should learn only $\sum\limits_{c \in [l]} q(\zeta_c)$. To hide this, we will add a polynomial $q_{blind}(\cdot)$ such that $\sum\limits_{c \in [l]} q_{blind}(\zeta_c) = 0$ and later $q_j(\eta_k)$ is opened to the verifier, hide this we need to add another polynomial to each $q_j(\cdot)$, say $q_{blind_j}(\cdot)$ such that $q_{blind_j}(\zeta_c) = 0$ $\forall c \in [l]$. To achieve this we will include one slice where each row but the $(m+1)th$ is encoding of zero vector. and in the $(m+1)th$ row is the encodig of a vector which sums to 0.

Similarly for quadratic also, we need to include one slice where each row is the encoding of zero vector.

10.2 Simulation based proof:

Let τ_1, τ_2, τ_3 be the transcripts of the interleaved check, linear check and quadratic check at the time of the execution of the protocol. where

$$\tau_1 = \{r_1, \tilde{c}, \gamma_1, w, Q_1, C_{Q_1}, \widetilde{U}_{Q_1}, \tilde{\delta}_{Q_1}, \tilde{v}_{Q_1}\}$$

$$\tau_2 = \{r_2, d_{1[h]}, q_1(\cdot), Q_2, \overline{q}_{1Q_2}, C_{Q_2}, U_{Q_2}, \delta_{Q_2}, v_{Q_2}\}$$

$$\tau_3 = \{r_3, d_{2[h]}, \gamma_2, q_2(\cdot), Q_2, \overline{q}_{2Q_2}, C_{Q_2}, U_{Q_2}, \delta_{Q_2}, v_{Q_2}\}$$

Let $\mathcal{S}$ be the simulator. Then $\mathcal{S}$ randomly picks $r_1, r_2, r_3, \gamma_1, \gamma_2, Q = Q_1 \cup Q_2$, where size of Q is t such that Q_1 and Q_2 are disjoint. $\mathcal{S}$ chooses a block $\mathcal{B}$ corresponding to Q, i.e. $\mathcal{S}$ chooses and fixes the complete matrix along the direction m and p, for the indices in Q. Once the block $\mathcal{B}$ is fixed, for these block, U is fixed by the condition that, t slices along the direction m and p is fixed and l positions are fixed because of the enconding of the witness. So, using $\mathcal{B}$, $\mathcal{S}$ generates $\tilde{c}, w, C_{Q_1}, \widetilde{U}_{Q_1}, d_{1[h]}, d_{2[h]}, \overline{q}_{1Q_2}, q_{2Q_2}, C_{Q_2}, U_{Q_2}$ and randomly picks $\tilde{\delta}_{Q_1}, \delta_{Q_2}, \tilde{v}_{Q_1}, v_{Q_2}$. Therefore the transcript generated by $\mathcal{S}$ is indistinguishable from τ_1, τ_2, τ_3.

11 Zero-Knowledge

We have three checks:

- Interleaved Testing
- Linear Check
- Quadratic Check

Note that during the interleaved testing, w is sent to the verifier, where w is the linear combination of all the rows of the encoding matrix, which verifier cannot compute on his own. So, it is required to blind the vector w by adding an additional vector which is a codeword.

For Linear check, verifier is receiving the polynomial $q(\cdot)$, and can evaluates in many points which he should not learn on his own, verifier should learn only $\sum\limits_{c \in [l]} q(\zeta_c)$. To hide this, we will add a polynomial $q_{blind}(\cdot)$ such that $\sum\limits_{c \in [l]} q_{blind}(\zeta_c) = 0$ and later $q_j(\eta_k)$ is opened to the verifier, hide this we need to add another polynomial to each $q_j(\cdot)$, say $q_{blind_j}(\cdot)$ such that $q_{blind_j}(\zeta_c) = 0 \ \forall c \in [l]$. To achieve this we will include one slice where each row but the $(m+1)th$ is encoding of zero vector. and in the $(m+1)th$ row is the encodig of a vector which sums to 0.

Similarly for quadratic also, we need to include one slice where each row is the encoding of zero vector.

11.1 Simulation for Zero-knowledge

Let τ_1, τ_2, τ_3 be the transcripts of the interleaved check, linear check and quadratic check at the time of the execution of the protocol. where

$$\tau_1 = \{r_1, \tilde{c}, \gamma_1, w, Q_1, C_{Q_1}, \widetilde{U}_{Q_1}, \tilde{\delta}_{Q_1}, \tilde{v}_{Q_1}\}$$

$$\tau_2 = \{r_2, d_{1[h]}, q_1(\cdot), Q_2, \overline{q}_{1Q_2}, C_{Q_2}, U_{Q_2}, \delta_{Q_2}, v_{Q_2}\}$$

$$\tau_3 = \{r_3, d_{2[h]}, \gamma_2, q_2(\cdot), Q_2, \overline{q}_{2Q_2}, C_{Q_2}, U_{Q_2}, \delta_{Q_2}, v_{Q_2}\}$$

But in our protocol we will be exectuing all the three protocols together and the final transcript will be:

$$\tau = \{r_1, r_2, r_3, \tilde{c}, d_{1[h]}, d_{2[h]}, \gamma_1, \gamma_2, w, q_1(\cdot), q_2(\cdot), Q_1, Q_2, \overline{q}_{1Q_2}, \overline{q}_{2Q_2}, C_Q, \widetilde{U}_{Q_1}, U_{Q_2}, \tilde{\delta}_{Q_1}, \delta_{Q_2}, \tilde{v}_{Q_1}, v_{Q_2}\}$$

Let $\mathcal{S}$ be the simulator. $\mathcal{S}$ picks $r_1, r_2, r_3, \gamma_1, \gamma_2$ unifromly at random.
Define: $Q_2' = \{k : (j,k) \in Q_2\}$. Let, $Q = Q_1 \cup Q_2'$, $\mathcal{S}$ picks Q uniformly at random. For our protocol, $|Q| = t$ and $Q_1 \cap Q_2' = \phi$.
Corresponding to Q, $\mathcal{S}$ chooses a block B i.e. $\{U[\cdot,\cdot,k] : k \in Q\}$ uniformly at random.
$\mathcal{S}$ computes the commitment of $U[i,\cdot,k]$, $\forall i \in [p], k \in Q$ and define as C_Q. The commitment is being done using the randomness δ_Q, chosen by $\mathcal{S}$ randomly.

$$C_{ik} = \mathsf{com}(U[i,\cdot,k], \delta_{ik}), \forall i \in [p], k \in Q$$

and define $\tilde{\delta}_{Q_1} = \sum_{i\in[p]} r_{1_i}\delta_{iQ_1}$. $\mathcal{S}$ generates $\tilde{c}$ in two parts $\tilde{c}_Q$ and $\tilde{c}_{\overline{Q}}$, where $\tilde{c}_Q = \{\tilde{c}_k : k \in Q\}$.
Define: $\tilde{c}_Q = \sum_{i\in[p]} r_{1_i} \cdot C_{iQ}$. Since $\tilde{c}$ is the commitment of $\widetilde{U}$, which should not reveal nothing about $\widetilde{U}$ and therefore picking remaining components $\tilde{c}$ i.e. $\tilde{c}_{\overline{Q}}$ has the uniform distribution over the set of all output of commitments.
Define $\widetilde{U}_{Q_1} = \sum_{i\in[p]} r_{1_i} \cdot U[i,\cdot,Q_1]$.
Now we will describe how the simulator $\mathcal{S}$ generates w, again it will be done in two parts: $w_Q, w_{\overline{Q}}$.
define $w_Q = \sum_{j\in[m]} [\gamma_{1_j} \cdot (\sum_{i\in[p]} r_{1_i} \cdot U[i,j,Q])] + \tilde{v}_Q$.
Let w' be the linear combination of the rows of U along the p and m direction without blinding. As w' is the encoding of the linear combination of the extended witness along the p and m direction and $\mathcal{S}$ fixed the block B of size t, which gives that $\exists$ only one w' which satisfies both the properties: i.e. $w'_k = \sum_{j\in[m]} [\gamma_{1_j} \cdot (\sum_{i\in[p]} r_{1_i} \cdot U[i,j,k])]$ $\forall k \in Q$ and w' is the encoding of $\sum_{j\in[m]} [\gamma_{1_j} \cdot (\sum_{i\in[p]} r_{1_i} \cdot x_{ijk})]$, if $\mathbf{x}$ is the extended witness. So w' has $t+l$ constraints, which fixes w'. Now for blinding $w = w'+v$, where v is a random codeword, of which $\mathcal{S}$ picks t components uniformly at random say, v_Q. Among the remaining components $\mathcal{S}$ can pick at most l many randomly so that v is a random codeword. So, $w = w' + v$ has the same distribution as v. So, for w from remaining components, $\mathcal{S}$ picks randomly l many of them as $w_{\overline{Q}} = w'_{\overline{Q}} + v_{\overline{Q}}$, has the same distribution as $v_{\overline{Q}}$.
$q_1(\cdot) = \sum_{j\in[m]} q_{1_j}(\cdot) + q_{1_{blind}}(\cdot)$ and $q_{1_j}(\cdot) = \sum_{i\in[p]} q_{1_{ij}}(\cdot) + q_{1_{blind_j}}(\cdot)$
$\mathcal{S}$ chooses a random polynomial $q_1(\cdot)$ of degree $< s+l-1$ such that $\sum_{c\in[l]} q_1(\zeta_c) = \overline{r}^T b$. From r_2, construct polynomials $r_{2_{ij}}(\cdot)$ of degree $< l$ such that $r_{2ij}(\zeta_k) = r_{2_{ijk}}$.
Set, $\overline{q}_1[k]_j = \sum_{i\in[p]} r_{2_{ij}}(\eta_k) \cdot U_{ijk} + U[p+1,j,k]$ $(j,k) \in Q_2$ this gives that $\overline{q}_{1Q_2}$ and for all remaining values of $(j,k) \in Q_2$, choose $\overline{q}_1[k]_j$ uniformly at random. Define: $d_{1[h]} = \mathsf{com}(\overline{q}_1[h])$.
Similarly we have, $q_2(\cdot) = \sum_{j\in[m]} \gamma_{3_j} \cdot q_{2_j}(\cdot) + q_{2_{blind}}(\cdot)$ and $q_{2_j}(\cdot) = \sum_{i\in[p]} r_{3_i} \cdot q_{2_{ij}}(\cdot) + q_{2_{blind_j}}(\cdot)$.
$\mathcal{S}$ picks a random polynomial $q_2(\cdot)$ of degree $< 2s-1$ such that $q_2(\zeta_c) = 0$ $\forall c \in [l]$. Now $\mathcal{S}$ needs to construct $\overline{q}_2$ and $d_{2[h]}$, for $(j,k) \in Q_2$, set $\overline{q}_2[k]_j = \sum_{i\in[p]} r_{3_i} \cdot U[i,j,k] + U[p+1,j,k]$, this gives $\overline{q}_{2Q_2}$ and for all remaining values of $(j,k) \in Q_2$, choose $\overline{q}_2[k]_j$ uniformly at random. Define: $d_{2[h]} = \mathsf{com}(\overline{q}_2[h])$.
We can see that $\mathcal{S}$ can generate a transcript which is indistinguishable from an actual transcript.

12 Amortized Verifier's complexity:

For our construction, verifier's complexity for Interleaved Testing and Quadratic Testing is already sublinear, main bottle-neck for verifier's complexity is due to linear check. Now we will discuss one by one, all the checks that verifier needs to do for the linearity constraint.

- Validation check for $C[k]$ with respect to c:
- $\sum_{k\in[l]} q(\zeta_k) = \overline{r}^T b$: since for the linear check in our protocol, we wilkl always have the public vecto $b = 0$, so only operations that the verifier will do is to evaluate the polynomial $q(\cdot)$ in l many public points, that needs $l \cdot (s_l - 1)$ many field operations, since degree of the polynomial $q(\cdot)$ is $< s + l - 1$.
- $\langle 1^m, \overline{q}[j'] \rangle = q(\eta_{j'})$ $\forall j' \in [h]$: For this proof of inner product will be given, we are using bulletproofs for that, as prover is giving multiple proofs, verifier will use the batch verification and verification complexity is: $\{2m + 2 + h(2\log(m) + 5)\}$ many exponentiation and $m \cdot h$ many multiplication, where $h = s + l - 1$.
- $\sum_{j'\in[h]} T_{j'k} d_{j'} = \overline{q}[k]$ $\forall k \in Q$: $t \cdot h$ many exponentiations
- $\langle e_j, U[i,.,k] \rangle = U[i,j,k]$ $\forall i \in [p], j \in Q', k \in Q$: $2m + 2 + t(\log m + 5)$ many exponentiations and $t \cdot m$ multiplications.

– $\sum_{i\in[p]} r_{ij}(\eta_k)\cdot U[i,j,k] = \overline{q}[k]_j \ \forall k \in Q, j \in Q'$: Time to compute $r_{ij}(\eta_k) + p\cdot t$ many multiplications. Now we will show that computating $r_{ij}(\cdot)$ will take $\max\{O(|C|^{2/c}), O(|C|^{1-1/c})\}$ on expectation.

12.1 Computing $r_{ij}(\eta_k)$

For this, the Verifier requires to read the whole matrix A, which of size $|C|$. So for a circuit the Verifier will a few computation and later use it to get better complexity.

We know that size of A is $pml \times pml$.
Let $r \in \mathbb{F}^{pml}$, then r can be written as $[r[1],\dots,r[p]]$, where

$$r[i] = \begin{bmatrix} r_{i11} & r_{i12} & \dots & r_{i1l} \\ r_{i21} & r_{i22} & \dots & r_{i2l} \\ \vdots & & & \\ r_{im1} & r_{im2} & \dots & r_{iml} \end{bmatrix}$$

Verifier needs to compute the evalutaions polynomials $r_{ij}(\eta_k) \ \forall i \in [p], j \in [m], k \in [n]$, where $r_{ij}(\cdot)$ is a polynomial of degree $< l$ such that $r_{ij}(\zeta_k) = r_{ijk} \ \forall i \in [p], j \in [m], k \in [l]$.
This above computation can be viewed as a linear transformation of r. Let T be the linear Transformation(Matrix), such that $r^T T = (r_{ij}(\eta_k))_{i,j,k}$. Then we can consider T as a matrix of dimension $pml \times pmn$ and as T transforms each row of r, which is of size l to a row of size n, so the structure of T is block diagonal, where T has pm many blocks with each each block of size $l \times n$.
A has at most 3 non-zero entries in each row.
Claim: $A \cdot T$ has at most $pml \times 3n$ non-zero entries.
Consider the i^{th} row of A, it has total $|C|$ many entries (including zeros and non-zeros), this can be partitioned into pm many portions where each portion is of length l. let us enumarate it as $s_1,\dots,s_{pm}$.
Now if any entry of s_j is non-zero, then due to that entry, j^{th} block out of the pm blocks of T will get multiplied and remain non-zero. So, one non-zero entry of of s_j can create n non-zero entries in the product matrix $A\cdot T$.
Now if all the non-zero entries of the i^{th} row of A are in different s_j's, therefore the product matrix $A\cdot T$ can have at most $3n$ many non-zero entries as i^{th} row of A can have at most 3 non-zero entries.
Let, B be the "*Bad*" parameter, i.e. If a column of $A\cdot T$ contains more than B non-zero entries call that column a "*Bad*" column. So, number of possible "*Bad*" column is $< \frac{3n|C|}{B}$.
Therefore probability that a randomly chosen column is "*Bad*" $= \frac{3n}{B}$.
So, for a randomly chosen column of $A\cdot T$, $A\cdot T[j]$, time to compute $r^T(A\cdot T[j])$, is dependent on the number of non-zero entries in $A\cdot T[j]$.
So if $A\cdot T[j]$ is a "*Bad*" column then the time is bounded above by $|C|$, otherwise i.e. $A\cdot T[j]$ is not "*Bad*", is bounded above by B.
So, expected time for the above computation is bounded by $\frac{3n}{B}\cdot|C| + (1-\frac{3n}{B})B$. Set, $B = |C|^{1-1/c}$, $n = |C|^{1/c}$
Expected computation $< \frac{3|C|^{1+1/c}}{|C|^{1-1/c}} + |C|^{1-1/c} - 3|C|^{1/c} = 3|C|^{2/c} + |C|^{1-1/c} - 3|C|^{1/c}$.
So for, $c > 2$ expected verifier's complexity is sublinear.