VeRange: Verification-efficient Zero-knowledge Range Arguments with Transparent Setup for Blockchain Applications and More

Yue Zhou
Australian National University

Sid Chi-Kin Chau*
CSIRO Data61

Abstract

Zero-knowledge range arguments are a fundamental cryptographic primitive that allows a prover to convince a verifier of the knowledge of a secret value lying within a predefined range. They have been utilized in diverse applications, such as confidential transactions, proofs of solvency and anonymous credentials. Range arguments with a transparent setup dispense with any trusted setup to eliminate security backdoor and enhance transparency. They are increasingly deployed in diverse decentralized applications on blockchains. One of the major concerns of practical deployment of range arguments on blockchains is the incurred gas cost and high computational overhead associated with blockchain miners. Hence, it is crucial to optimize the *verification efficiency* in range arguments to alleviate the deployment cost on blockchains and other decentralized platforms. In this paper, we present VeRange with several new zero-knowledge range arguments in the discrete logarithm setting, requiring only $c\sqrt{N/\log N}$ group exponentiations for verification, where N is the number of bits to represent a range and c is a small constant, making them concretely efficient for blockchain deployment with a very low gas cost. Furthermore, VeRange is aggregable, allowing a prover to simultaneously prove T range arguments in a single argument, requiring only $O(\sqrt{TN/\log(TN)})+T$ group exponentiations for verification. We deployed VeRange on Ethereum and measured the empirical gas cost, achieving the fastest verification runtime and the lowest gas cost among the discrete-logarithm-based range arguments in practice.[1]

Keywords

Zero-knowledge range argument, transparent setup, blockchain

1 Introduction

In modern cryptographic protocols, it is often that one party has to commit to a secret value (or a witness to a statement) in a cryptographic commitment and is required to subsequently reveal a certain property of the secret value to another party, without completely disclosing it. One of the most fundamental properties in real-world applications is the property of a value lying within a given range. A *zero-knowledge range argument* allows a prover to convince a verifier that the committed value ω in a prior commitment $\mathtt{Cm}(\omega)$ is within $[0, 2^N\text{-}1]$, without revealing ω. In this paper, we consider a range given in form of $[0, 2^N\text{-}1]$, which can be generalized to a general range $[\underline{\omega}, \overline{\omega}]$ straightforwardly.

There is a growing list of real-world scenarios for zero-knowledge range arguments in practice:

(1) ***Confidential Transactions***: Typical cryptocurrencies (e.g., Bitcoin, Ethereum) feature a public ledger that exposes the transaction records to the public. To avoid public scrutiny, one can hide the account balances in cryptographic commitments [22]. To ensure correctness and consistency, a transaction request must attest that the sum of the output amounts does not exceed the input amounts, namely, the net transfer is not a deficit.
(2) ***Proofs of Solvency***: Nowadays cryptocurrency exchanges are required by regulators to certify that the amount in reserves should be able to cover the amount in liabilities (including payouts to customers), and hence, maintaining solvency [16]. It is desirable that the proofs of solvency do not reveal any holding accounts, which may become a target for attacks.
(3) ***e-Voting***: Electronic votes can be cast to an election authority in a privacy-preserving manner by encryption. At the end of the election, the votes are tallied by homomorphic encryption and decrypted by the electoral commission [23]. To ensure its correctness, each encrypted vote is verified for its positivity.
(4) ***Anonymous Credentials***: Sensitive personal attributes, such as date of birth and income, should be protected securely [9]. Often, some proofs of sensitive personal attributes (e.g., age, income) meeting certain criteria thresholds are required to gain access (e.g., for liquor) or privilege (e.g., social welfare).
(5) ***Verifiable Auctions***: In a second-price auction, the winning bidder pays the second-highest price of all bids. In a verifiable auction [1], the auctioneer provides a proof of the second-highest price of bids, without revealing the prices of all other bids (including the winning bid).
(6) ***Data Sovereignty***: In many data-driven applications (e.g., federated learning [2]), distributed parties coordinate each other by a protocol to utilize local data. To ensure protocol adherence, each party proves certain properties of their data (e.g., within a certain range) without sacrificing data privacy.

Furthermore, decentralized applications are increasingly popular, which operate on blockchain platforms and are executed by publicly verifiable smart contracts via a network of distributed miners. Many of the above scenarios are relevant to the context of decentralized applications, such as confidential transactions, proofs of solvency, e-voting, anonymous credentials. For decentralized applications, a trustless setting without a trusted setup is crucial. Although a trusted setup can be established by multi-party computation [7, 24], there is no publicly verifiable mechanism to eliminate collusion among the setup parties, particularly for blockchains. Hence, this paper focuses on range arguments with a *transparent setup* that enhances transparency and enables trustless decentralized applications without entrusting to any third-party for setup.

On a blockchain platform (e.g., Ethereum), distributed miners usually replicate the execution of smart contracts independently to

*Corresponding author: sid.chau@acm.org
[1]This is an extended version of the conference paper in AsiaCCS'25 [35].

achieve global consistency. As a result, miners charge cryptocurrencies (so-called gas fees) to smart contract invokers per smart contract execution for the incurred resources (i.e, computation and memory storage). Gas cost is used to measure the amount of computational resources to execute the required operations in a smart contract. One of the major concerns of practical deployment of range arguments is the incurred gas cost and high computational overhead associated with miners. Based on our measurements of empirical gas cost over Ethereum, we observe that the majority of gas cost of the existing range arguments is attributed to the computational tasks, rather than the memory storage. Traditional range arguments (e.g., Bulletproofs) can incur over USD$100 gas fee on Ethereum for the verification of a 128-bit range (see an empirical study in Sec. 6). Hence, it is crucial to optimize the *verification efficiency* in range arguments to alleviate the deployment cost on blockchains and other decentralized platforms. In this paper, we devise practically efficient zero-knowledge range arguments with a transparent setup and a very low gas cost on blockchains.

1.1 Related Work

We survey the zero-knowledge range arguments in the literature:

(1) *Zero-knowledge Set Membership*: This class of zero-knowledge range arguments prove a value lying in an arbitrary set [9]. Every element in the set is associated with a signature published by the verifier. A prover can prove the set membership by a proof of knowledge of a signature of an element. A drawback of this approach is the signatures for a range scaling linearly with its size, making it very inefficient for a large range.

(2) *Four-square Decomposition*: Lagrange's four square theorem states that every integer can be decomposed into a sum of squares of integers. Early range arguments of this class were based on integer commitments of unknown order groups [6], which require a trusted setup or ideal class group[2]. Recent bounded integer commitments [14, 15] follow a weaker soundness model (called "relaxed soundness") and hence cannot be applied to confidential transactions on blockchains. Although [14] proposes a way to strengthen relaxed soundness, this still requires a trusted setup or ideal class group.

(3) *Hash Chains*: A hash chain is sequential evaluation of a hash function on an unknown random input for x times, which can be regarded as a commitment on x. Hash chains were utilized in micropayments [29] and location hiding [10]. But hash chains are not homomorphic commitments, which are unsuitable for confidential transactions and e-voting.

(4) *Binary/B-ary Digital Decomposition*: If $\omega \in [0, 2^N - 1]$, then ω can be expressed by bit decomposition as $\omega = \sum_{i\in[N]} b_i \cdot 2^{i-1}$, where $b_i \in \{0, 1\}$. In general, if $\omega \in [0, B^{\tilde{N}} - 1]$, then ω can be expressed by $\omega = \sum_{i\in[\tilde{N}]} d_i \cdot B^{i-1}$, where $d_i \in \{0, 1, ..., B-1\}$. The range arguments based on binary/B-ary digital decomposition validate the satisfiability of witness b_i (or d_i) in a constraint system. Although one can apply general zk-SNARKs (e.g., [4, 21]) to automate the generation of zero-knowledge range arguments via general arithmetic circuits, this approach either requires a trusted setup, or results in inefficient range arguments because of the overhead of translation from general arithmetic circuits. Hence, we optimize the efficiency to a large extent by designing specific zero-knowledge proofs for range arguments.

See [12] for a recent survey of zero-knowledge range arguments.

Table 1: A comparison of range arguments with transparent setup in discrete logarithm settings.

Scheme	Proof Size	Verification Time	Proving Time
Bulletproofs [8]	$2\log N \ \lvert\mathbb{G}\rvert$	$2N + 2\log N$ $\mathbb{G}$ Exp	$9N + 4\log N$ $\mathbb{G}$ Exp
Bulletproofs+ [13]	$2\log N \ \lvert\mathbb{G}\rvert$	$2N + 2\log N$ $\mathbb{G}$ Exp	$9N + 4\log N$ $\mathbb{G}$ Exp
SwiftRange [31]	$4\log N \ \lvert\mathbb{G}\rvert$	$N + 4\log N$ $\mathbb{G}$ Exp	$8N$ $\mathbb{G}$ Exp
Bulletproofs++[18]	$O(\log(\frac{N}{\log N}))\ \lvert\mathbb{G}\rvert$	$O(\frac{N}{\log N})$ $\mathbb{G}$ Exp	$O(\frac{N}{\log N})$ $\mathbb{G}$ Exp
Flashproofs [30]	$\frac{N^{2/3}+3N^{1/3}}{2}\ \lvert\mathbb{G}\rvert + N^{2/3}\ \lvert\mathbb{Z}_p\rvert$	$\frac{3(N^{2/3}+N^{1/3})}{2}$ $\mathbb{G}$ Exp	$\frac{N^{4/3}+N+3N^{2/3}+5N^{1/3}}{2}$ $\mathbb{G}$ Exp
BG18 [5]	$O(\frac{N^2}{\log N})\ \lvert\mathbb{G}\rvert + O(\frac{N}{\log N})\ \lvert\mathbb{Z}_p\rvert$	$O(N)$ $\mathbb{G}$ Exp	$O(N)$ $\mathbb{G}$ Exp
BCCGP[a] [4]	$O(\sqrt{N})\ \lvert\mathbb{G}\rvert + O(\sqrt{N})\ \lvert\mathbb{Z}_p\rvert$	$O(\sqrt{N})$ $\mathbb{G}$ Exp	$O(N)$ $\mathbb{G}$ Exp
BFGW20[b] [3]	$O(\log N)\ \lvert\mathbb{G}_U\rvert$	$O(\log N)$ $\mathbb{G}_U$ Exp	$O(N \log N)$ $\mathbb{G}_U$ Exp
LLRing[c] [25]	$6\log N\ \lvert\mathbb{G}_T\rvert$	$9\log N$ $\mathbb{G}_T$ Exp (PreComp: $2N$ $\mathbb{P}$ + N $\mathbb{G}$ Exp)	$10N$ $\mathbb{P}$ + $4N$ $\mathbb{G}$ Exp
VeRange Type-1	$2N^{1/2}\ \lvert\mathbb{G}\rvert + N\ \lvert\mathbb{Z}_p\rvert$	$3N^{1/2}$ $\mathbb{G}$ Exp	$N + 4N^{1/2}$ $\mathbb{G}$ Exp
VeRange Type-2	$5.2(\frac{N}{\log N})^{1/2}\ \lvert\mathbb{G}\rvert + 4\frac{N}{\log N}\ \lvert\mathbb{Z}_p\rvert$	$6.7(\frac{N}{\log N})^{1/2}$ $\mathbb{G}$ Exp	$2\frac{N}{\log N}$ + $10.5(\frac{N}{\log N})^{1/2}$ $\mathbb{G}$ Exp
VeRange Type-2B	$1.7(\frac{N}{\log N})^{2/3} + 2.9(\frac{N}{\log N})^{1/3}\ \lvert\mathbb{G}\rvert + 2.6(\frac{N}{\log N})^{2/3}\ \lvert\mathbb{Z}_p\rvert$	$3.8(\frac{N}{\log N})^{2/3} + 1.7(\frac{N}{\log N})^{1/3}$ $\mathbb{G}$ Exp	$8.9(\frac{N}{\log N})^{2/3} + 2.9(\frac{N}{\log N})^{1/3}$ $\mathbb{G}$ Exp
VeRange Type-3	$4.2(\frac{N}{\log N})^{1/2}\ \lvert\mathbb{G}\rvert + 2.8(\frac{N}{\log N})^{1/2}\ \lvert\mathbb{Z}_p\rvert$	$6.6(\frac{N}{\log N})^{1/2}$ $\mathbb{G}$ Exp	$3.4\frac{N}{\log N} + 7.1(\frac{N}{\log N})^{1/2}$ $\mathbb{G}$ Exp

Note: In our performance estimation, we only state the most significant terms. $\lvert\mathbb{G}\rvert$ means group elements, $\lvert\mathbb{Z}_p\rvert$ means field elements, $\mathbb{G}$ Exp means group exponentiations. [a]BCCGP scheme can generate unoptimized range arguments from arithmetic circuits. [b]BFGW20 requires an unknown order group ($\mathbb{G}_U$) to yield logarithmic verification time. [c]LLRing requires precomputation and pairing operations ($\mathbb{P}$) on a pairing-friendly target group ($\mathbb{G}_T$) to yield logarithmic verifiability.

In this paper, we focus on *zero-knowledge proofs specifically optimized for range arguments*, which yield the most efficient and cost-effective solutions for practical deployments. Since range arguments are a fundamental primitive, it is worthwhile to optimize their efficiency for many applications.

In Table 1, we compare with the existing range arguments with transparent setup in discrete logarithm settings based on specific zero-knowledge proof systems. We omit other studies that rely on a trusted setup to enable efficient range arguments (e.g., [27]).

Bulletproofs [8] is a general proof system for inner-product relations with logarithmic proof size and linear verification time based on a recursive folding technique. Bulletproofs includes a specific proof system for range arguments based on binary decomposition that takes around $2N$ group exponentiations for verification. Bulletproofs+ [13] makes slight improvement over the verification time. Bulletproofs++ [18] extends Bulletproofs to B-ary digital decomposition based on a reciprocal relation, which results in $O(\frac{N}{\log N})$ group exponentiations. But their approach relies on arithmetic circuits, which is not optimized specifically for range arguments. Recently, SwiftRange [31] improves Bulletproofs by halving the group exponentiations to N at the expense of doubling the proof size. Dory [26] improves upon Bulletproofs' recursive folding technique by leveraging precomputation and pairing, which yields logarithmic verification time. LLRing [25] developed a logarithmic linkable ring scheme based on Dory. Their technique also applies to range arguments to give logarithmic verifiable range arguments. But Dory uses pairing, which is not as concretely efficient as the basic discrete logarithm setting and the empirical runtime is far higher than Bulletproofs for typical ranges (see our comparison in Sec. 6). BG18

[2]Unknown order group by ideal class group with a transparent setup is not yet concretely efficient to implement for practical applications. For example, at the 128-bit security level, [17] suggests that $\approx$ 6656-bit is required. There is a performance comparison between unknown order groups and pairing-friendly groups in [26]. The state-of-the-art practical implementation of unknown order groups using ideal class group takes 27000µs for a group multiplication, whereas the same study shows that a finite group multiplication takes only 42µs. Moreover, there is no existing class group implementation on today's blockchain platforms (e.g., Ethereum). Hence, it is not practical to use unknown order groups with a transparent setup on blockchains.

[5] and BFGW20 [3] present specific proof systems for range arguments. BG18 is designed for batch verification, and takes linear verification time. BFGW20 yields logarithmic verification time but needs an unknown order group that requires a trusted setup or ideal class groups. The most practically verification-efficient range argument so far is Flashproofs [30] (range argument) that yields $O(N^{2/3})$ group exponentiations and is empirically measured to have a low gas cost on Ethereum.

1.2 Our Contributions

In this paper, we devise VeRange, consisting of three types of verification-efficient zero-knowledge range arguments with transparent setup in the discrete logarithm setting. VeRange contains multiple types of range arguments, two of which require only $c\sqrt{\frac{N}{\log N}}$ group exponentiations for verification, where $c < 7$ is a small constant, making them concretely efficient for blockchain deployment with a very low gas cost. Although one can apply general zk-SNARKs (e.g., [4]) to an arithmetic circuit representing binary decomposition of a number to achieve $O(\sqrt{N})$ verification time, this results in unoptimized range arguments that are not as concretely efficient as our specific range argument systems.

In many applications, a single prover needs to prove multiple range arguments simultaneously. For example, a confidential transaction contains multiple output amounts. In proofs of solvency, a cryptocurrency exchange needs to show range arguments for every holding account. Given the sub-linear verification time and proof size of VeRange, we aggregate T arguments that is more efficient than verifying T individual arguments. We provide *aggregable* VeRange, two of which require only $O(\sqrt{\frac{TN}{\log(TN)}}) + T$ group exponentiations in the verification of T arguments together.

We deployed VeRange on Ethereum and measured the empirical gas cost, achieving the fastest verification runtime and the lowest gas cost among the discrete-logarithm-based range arguments in practice, particularly for aggregating multiple range arguments.

Table 1 compares the theoretical performance of VeRange with the extant range arguments of transparent setup. VeRange attains the lowest number of group exponentiations for verification among the range arguments without pairing or unknown order groups.

In the following, we outline each type of VeRange and its novelty:

- VeRange Type-1: We optimize Flashproofs to reduce from $O(N^{2/3})$ group exponentiations for verification to only $3N^{1/2}$ at the expense of a linear number of field elements in the proof. It achieves the fastest verification time in the recent discrete logarithm setting with a transparent setup.
- VeRange Type-2: We utilize the reciprocal relation from Bulletproofs++ to realize B-ary digital decomposition with $O((\frac{N}{\log N})^{1/2})$ group exponentiations for verification. It achieves the fastest proving time in the recent discrete logarithm setting. We also develop VeRange type-2B that combines the ideas of Flashproofs and Bulletproofs++ to yield a lower gas-cost alternative at the expense of $O((\frac{N}{\log N})^{2/3})$ group exponentiations for verification.
- VeRange Type-3: We design an efficient range argument based on efficient batch verification of polynomial evaluation. Although our approach is based on BG18 [5], our approach differs from BG18, as we optimize batch verification to reduce group exponentiations for verification from $O(N)$ to $O((\frac{N}{\log N})^{1/2})$. It achieves the lowest gas cost with a transparent setup.

Paper Organization. Sec. 2 presents the preliminaries. Secs. 3-5 present the three types of VeRange, respectively. Sec. 6 empirically evaluates the performance of VeRange on Ethereum and compares with the extant range arguments. Due to the page limit, some definitions and technical proofs are deferred to the Appendix.

2 Preliminaries

In this section, we present the preliminaries and definitions for our work. Let λ be the security level parameter and $\mathrm{negl}(\lambda)$ be a negligible function of λ. PPT denotes "probabilistic polynomial time". "$\xleftarrow{\$}$" denotes a uniformly random selection from a set.

Vectors. Denote a cyclic group of prime order p by $\mathbb{G}$, and a ring of integers modulo p by $\mathbb{Z}_p$. Let $\mathbb{Z}_p^* \triangleq \mathbb{Z}_p \backslash \{0\}$. Denote a vector in bold font with an arrow symbol and its coordinates in normal font with subscripts (e.g., $\vec{\mathbf{a}} \triangleq (a_1, ..., a_n) \in \mathbb{Z}_p^n$ denotes a vector of field elements and $\vec{\mathbf{G}} \triangleq (G_1, ..., G_n) \in \mathbb{G}^n$ denotes a vector of group generators).

Commitment Scheme. A commitment scheme is a mapping $\mathtt{Cm} : \mathcal{M}^n \times \mathcal{R} \to \mathcal{C}$ from a (vector) message space $\mathcal{M}^n$ and a random mask space $\mathcal{R}$ to a commitment space $\mathcal{C}$. A commitment scheme is *homomorphic*, if for any $\vec{\mathbf{m}}_1, \vec{\mathbf{m}}_2 \in \mathcal{M}^n, \mathsf{r}_1, \mathsf{r}_2 \in \mathcal{R}$:

$$\mathtt{Cm}(\vec{\mathbf{m}}_1; \mathsf{r}_1) \cdot \mathtt{Cm}(\vec{\mathbf{m}}_2; \mathsf{r}_2) = \mathtt{Cm}(\vec{\mathbf{m}}_1 + \vec{\mathbf{m}}_2; \mathsf{r}_1 + \mathsf{r}_2)$$

Pedersen commitment scheme is a homomorphic commitment scheme that is perfectly hiding and computationally binding.

Definition 2.1 (Pedersen Commitment). Let $\mathcal{M} = \mathbb{Z}_p^n$, $\mathcal{R} = \mathbb{Z}_p^*$ and $\mathcal{C} = \mathbb{G}$ of order p. Let $\vec{\mathbf{G}} \xleftarrow{\$} \mathbb{G}^n, Q \xleftarrow{\$} \mathbb{G}$ be randomly selected generators. Define Pedersen commitment by

$$\mathtt{Cm}(\vec{\mathbf{m}}; \mathsf{r}) \triangleq \vec{\mathbf{G}}^{\vec{\mathbf{m}}} \cdot Q^{\mathsf{r}} = \Big(\prod_{i \in [n]} G_i^{\mathsf{m}_i} \Big) \cdot Q^{\mathsf{r}}$$

Definition 2.2 (Discrete Logarithm Relation (DLR)). The DLR assumption holds for any PPT adversary $\mathcal{A}$ for a given η:

$$\Pr\left[\begin{array}{c} \vec{\mathbf{x}} \leftarrow \mathcal{A}[\vec{\mathbf{G}}], \\ \vec{\mathbf{G}}^{\vec{\mathbf{x}}} = \eta \end{array} \middle| \begin{array}{c} \mathbb{G} \leftarrow \mathtt{Setup}[1^\lambda], \\ \vec{\mathbf{G}} \xleftarrow{\$} \mathbb{G} \end{array} \right] \leq \mathrm{negl}(\lambda)$$

Namely, non-trivial discrete logarithm relations among random generators $\vec{\mathbf{G}}$ cannot be discovered by a PPT adversary.

Zero-Knowledge Arguments of Knowledge. An *argument system* is consisted of three PPT algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$, where $\mathcal{G}$ is the setup algorithm for public parameters pp, $\mathcal{P}$ and $\mathcal{V}$ are the prover and verifier algorithms. Denote the communication transcript between the prover and verifier by $\mathsf{tr} \leftarrow \langle \mathcal{P}(\cdot), \mathcal{V}(\cdot) \rangle$. At the end, the transcript will produce a binary decision: $\mathtt{Accept}[\mathsf{tr}] \in \{0, 1\}$. Range arguments belong to *zero-knowledge arguments of knowledge*. See Appendix B for more detailed definitions.

Definition 2.3 (Argument of Knowledge). Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is an *argument of knowledge* for a relation, if it satisfies perfect *completeness* (Def. (B.2)) and *computational witness-extended emulation* (CWE) (Def. (B.3)).

Essentially, CWE captures the idea of *knowledge-sound* arguments. Informally, if an adversary produces an acceptable argument with some probability, there exists an emulator that produces a similar argument and a witness with the same probability.

We are interested in *Special Honest-Verifier Zero-Knowledge (SHVZK)* arguments (Def. (B.5)) that do not leak the information of the witness beyond what can be inferred from the truth of the statement.

Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is called *public-coin* (Def. (B.4)), if the verifier chooses her messages uniformly at random, independent from the messages sent by the prover.

In this paper, we focus on *multi-move interactive public-coin protocols* for arguments of knowledge. The Fiat-Shamir transformation can be applied to convert interactive protocols to non-interactive arguments using the random oracle model by replacing the public-coin challenges by the output of a cryptographic hash function [20]. This reduces multiple moves in an interactive protocol to a single move in a publicly verifiable scheme.

3 VeRange Type-1 Range Argument

In this section, we introduce the type-1 range argument. First, we describe the main ideas at a high level, before presenting the range argument protocol and aggregated range argument.

3.1 Technical Overview

If $\omega \in [0, 2^N - 1]$, we can write $\omega = \sum_{i\in[N]} b_i \cdot 2^{i-1}$ by bit decomposition, where $b_i \in \{0, 1\}$. Hence, one can check if $\omega \in [0, 2^N - 1]$ by checking if there exists a vector $\vec{\mathbf{b}} = (b_1, ..., b_N) \in \mathbb{Z}_p^N$, such that

$$\begin{cases} b_i(1 - b_i) \stackrel{?}{=} 0, \forall i \in [N] \\ \sum_{i\in[N]} b_i \cdot 2^{i-1} \stackrel{?}{=} \omega \end{cases} \tag{1}$$

We next outline the basic idea of an efficient zero-knowledge proof protocol that checks Eqn. (1) with respect to a (Pedersen) scalar commitment of ω, i.e., $\mathtt{Cm}(\omega) \triangleq G^{\omega} \cdot Q^{\mathsf{r}_\omega}$, where $\mathsf{r}_\omega \xleftarrow{\$} \mathbb{Z}_p^*$ is a random mask, without revealing $\vec{\mathbf{b}}$.

We arrange $\vec{\mathbf{b}}$ in a $J \times K$ matrix $\left(\hat{b}_{j,k}\right)_{j=1,k=1}^{J\ K}$, as defined by

$$\hat{b}_{j,k} \triangleq \begin{cases} b_{J(k-1)+j}, & \text{if } J(k-1)+j \le N \\ 0, & \text{if } J(k-1)+j > N \end{cases}$$

Also, define a $J \times K$ matrix $\left(\hat{2}_{j,k}\right)_{j=1,k=1}^{J\ K}$ by

$$\hat{2}_{j,k} \triangleq \begin{cases} 2^{J(k-1)+j-1}, & \text{if } J(k-1)+j \le N \\ 0, & \text{if } J(k-1)+j > N \end{cases}$$

Then, define $\left(w_{j,k} \triangleq \hat{b}_{j,k} \cdot \hat{2}_{j,k}\right)_{j=1,k=1}^{J\ K}$, namely,

$$\begin{pmatrix} w_{1,1} & \dots & w_{1,K} \\ \vdots & \ddots & \vdots \\ w_{J,1} & \dots & w_{J,K} \end{pmatrix} \triangleq \begin{pmatrix} b_1 \cdot 2^0 & b_{J+1} \cdot 2^J & \dots & b_{J(K-1)+1} \cdot 2^{J(K-1)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{J-\eta} \cdot 2^{J-\eta-1} & b_{2J-\eta} \cdot 2^{2J-\eta-1} & \dots & b_N \cdot 2^{N-1} \\ b_{J-\eta+1} \cdot 2^{J-\eta} & b_{2J-\eta+1} \cdot 2^{2J-\eta} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_J \cdot 2^{J-1} & b_{2J} \cdot 2^{2J-1} & \dots & 0 \end{pmatrix}$$

where $\eta = JK \bmod N$, such that $\eta < J$. Hence, Eqn. (1) becomes

$$\begin{cases} w_{j,k}(\hat{2}_{j,k} - w_{j,k}) \stackrel{?}{=} 0, \forall j \in [J], k \in [K] \\ \sum_{j\in[J],k\in[K]} w_{j,k} \stackrel{?}{=} \omega \end{cases} \tag{2}$$

In this range argument, the prover first commits $(\sum_{j\in[J]} w_{j,k})_{k\in[K]}$ as $(W_k \triangleq G^{\sum_{j\in[J]} w_{j,k}} \cdot Q^{\mathsf{r}_k^{(W)}})_{k\in[K]}$, where the random masks $(\mathsf{r}_k^{(W)})_{k\in[K]}$ are set to satisfy $\mathsf{r}_\omega = \sum_{k\in[K]} \mathsf{r}_k^{(W)}$. Then, the verifier can check if $\omega \in [0, 2^N - 1]$ by checking

$$\mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[K]} W_k \text{ and } w_{j,k} \stackrel{?}{\in} \{0, \hat{2}_{j,k}\},\ \forall k \in [K], j \in [J] \tag{3}$$

Next, we proceed to check the satisfiability of $(w_{j,k} \stackrel{?}{\in} \{0, \hat{2}_{j,k}\})_{j\in[J],k\in[K]}$ by the below zero-knowledge protocol:

1. In addition to $(W_k)_{k\in[K]}$, the prover also commits $\left((T_k)_{k\in[K]}, S, R\right)$, which will be defined in the following.
2. The verifier then sends a random challenge vector $\vec{\epsilon} \xleftarrow{\$} \mathbb{Z}_p^{*K}$ to the prover.
3. The prover replies with $(v_{j,k} \triangleq w_{j,k} \cdot \epsilon_k + \mathsf{r}_{j,k})_{j\in[J],k\in[K]}$, where $\mathsf{r}_{j,k} \xleftarrow{\$} \mathbb{Z}_p^*$ is a random mask.
4. The verifier then computes $(u_{j,k} \triangleq \hat{2}_{j,k} \cdot \epsilon_k - v_{j,k})_{j\in[J],k\in[K]}$.
5. Next, the satisfiability of $(w_{j,k})_{j\in[J],k\in[K]}$ can be checked by the following:

$$\sum_{k\in[K]} v_{j,k} \cdot u_{j,k} \stackrel{?}{=} \sum_{k\in[K]} \underbrace{(\hat{2}_{j,k} - w_{j,k}) w_{j,k}}_{=0 \text{ if } w_{j,k}\in\{0,\hat{2}_{j,k}\}} \cdot \epsilon_k^2 + \sum_{k\in[K]} \mathsf{r}_{j,k}(\hat{2}_{j,k} - 2w_{j,k}) \cdot \epsilon_k - \sum_{k\in[K]} (\mathsf{r}_{j,k})^2$$

   By the DLR assumption, one can equivalently check the following equation:

$$\prod_{j\in[J]} H_j^{\sum_{k\in[K]} v_{j,k}\cdot u_{j,k}} \cdot Q^{\eta_1} \stackrel{?}{=} \prod_{k\in[K]} T_k^{\epsilon_k} \cdot S \tag{4}$$

   where $(T_k \triangleq \prod_{j\in[J]} H_j^{\mathsf{r}_{j,k}(\hat{2}_{j,k} - 2w_{j,k})} \cdot Q^{\mathsf{r}_k^{(T)}})_{k\in[K]}$ and $S \triangleq \prod_{j\in[J]} H_j^{-\sum_{k\in[K]}(\mathsf{r}_{j,k})^2} \cdot Q^{\mathsf{r}_S}$ should be committed by the prover at (1) before knowing $\vec{\epsilon}$ and the prover provides $\eta_1 \triangleq \vec{\mathsf{r}}^{(T)} \cdot \vec{\epsilon} + \mathsf{r}_S$.
6. To relate $(v_{j,k})_{j\in[J],k\in[K]}$ and $(W_k)_{k\in[K]}$, one can check the following:

$$\sum_{k\in[K]} \sum_{j\in[J]} v_{j,k} \stackrel{?}{=} \sum_{k\in[K]} \left(\epsilon_k \cdot \sum_{j\in[J]} w_{j,k} + \sum_{j\in[J]} \mathsf{r}_{j,k}\right) \tag{5}$$

   By the DLR assumption, one can equivalently check the following equation:

$$G^{\sum_{j\in[J],k\in[K]} v_{j,k}} \cdot Q^{\eta_2} \stackrel{?}{=} \prod_{k\in[K]} W_k^{\epsilon_k} \cdot R \tag{6}$$

   where $R \triangleq G^{\sum_{j\in[J],k\in[K]} \mathsf{r}_{j,k}} \cdot Q^{\mathsf{r}_R}$ should be committed by the prover at (1) before knowing $\vec{\epsilon}$ and the prover provides $\eta_2 \triangleq \vec{\mathsf{r}}^{(W)} \cdot \vec{\epsilon} + \mathsf{r}_R$.

Figure 1: VeRange **type-1 range argument protocol**

$$\Pi_{\text{ty1}}\left[\mathtt{Cm}(\omega) \in \mathbb{G};\ \omega \in \mathbb{Z}_p, \mathsf{r}_\omega \in \mathbb{Z}_p^*\right]$$

$$\mathcal{P}: \vec{\mathbf{b}} \in \{0,1\}^N \text{ is the bit-decomposition of } \omega \text{ such that } \omega = \sum_{i\in[N]} b_i \cdot 2^{i-1}$$

$$\left(\mathsf{r}_{j,k} \xleftarrow{\$} \mathbb{Z}_p^*\right)_{j\in[J],k\in[K]},\quad \vec{\mathsf{r}}^{(W)}, \vec{\mathsf{r}}^{(T)} \xleftarrow{\$} \mathbb{Z}_p^{*K},\quad \mathsf{r}_R, \mathsf{r}_S \xleftarrow{\$} \mathbb{Z}_p^*,$$

$$\mathsf{r}_K^{(W)} \triangleq \mathsf{r}_\omega - \sum_{k\in[K-1]} \mathsf{r}_k^{(W)},\quad \left(w_{j,k} \triangleq \hat{b}_{j,k} \cdot \hat{2}_{j,k}\right)_{j\in[J],k\in[K]}$$

$$\left(t_{j,k} \triangleq \mathsf{r}_{j,k} \cdot (\hat{2}_{j,k} - 2w_{j,k})\right)_{j\in[J],k\in[K]}$$

$$\textcircled{1}\ \mathcal{P} \Rightarrow \mathcal{V}: \left(W_k \triangleq G^{\sum_{j\in[J]} w_{j,k}} \cdot Q^{\mathsf{r}_k^{(W)}},\quad T_k \triangleq \prod_{j\in[J]} H_j^{t_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\right)_{k\in[K]} \tag{7}$$

$$R \triangleq G^{\sum_{j\in[J],k\in[K]} \mathsf{r}_{j,k}} \cdot Q^{\mathsf{r}_R},\quad S \triangleq \prod_{j\in[J]} H_j^{-\sum_{k\in[K]} (\mathsf{r}_{j,k})^2} \cdot Q^{\mathsf{r}_S} \tag{8}$$

$$\textcircled{2}\ \mathcal{P} \Leftarrow \mathcal{V}: \vec{e} \xleftarrow{\$} \mathbb{Z}_p^{*K} \tag{9}$$

$$\textcircled{3}\ \mathcal{P} \Rightarrow \mathcal{V}: \left(v_{j,k} \triangleq w_{j,k} \cdot \epsilon_k + \mathsf{r}_{j,k}\right)_{j\in[J],k\in[K]},\ \eta_1 \triangleq \vec{\mathsf{r}}^{(T)} \cdot \vec{e} + \mathsf{r}_S,\ \eta_2 \triangleq \vec{\mathsf{r}}^{(W)} \cdot \vec{e} + \mathsf{r}_R \tag{10}$$

$$\textcircled{4}\quad \mathcal{V}: \left(u_{j,k} \triangleq \hat{2}_{j,k} \cdot \epsilon_k - v_{j,k}\right)_{j\in[J],k\in[K]}$$

$$\textsc{Check} \begin{cases} \textcircled{5} & \prod_{j\in[J]} H_j^{\sum_{k\in[K]} v_{j,k} \cdot u_{j,k}} \cdot Q^{\eta_1} \overset{?}{=} \prod_{k\in[K]} T_k^{\epsilon_k} \cdot S \\ \textcircled{6} & G^{\sum_{j\in[J],k\in[K]} v_{j,k}} \cdot Q^{\eta_2} \overset{?}{=} \prod_{k\in[K]} W_k^{\epsilon_k} \cdot R \\ & \mathtt{Cm}(\omega) \overset{?}{=} \prod_{k\in[K]} W_k \end{cases} \tag{11}$$

Figure 2: Aggregated VeRange **type-1 range argument protocol**

$$\Pi_{\text{a.ty1}}\left[\left(\mathtt{Cm}(\omega^{(t)}) \in \mathbb{G}\right)_{t\in[T]};\ \left(\omega^{(t)} \in \mathbb{Z}_p, \mathsf{r}_{\omega(t)} \in \mathbb{Z}_p^*\right)_{t\in[T]}\right]$$

$$\mathcal{P} \Leftarrow \mathcal{V}: \gamma \xleftarrow{\$} \mathbb{Z}_p^*$$

$$\mathcal{P}: \vec{\mathbf{b}}^{(t)} \in \{0,1\}^N \text{ is the bit-decomposition of } \omega^{(t)} \text{ such that } \omega^{(t)} = \sum_{i\in[N]} b_i^{(t)} \cdot 2^{i-1}$$

$$\left(\mathsf{r}_{j,k} \xleftarrow{\$} \mathbb{Z}_p^*\right)_{j\in[J],k\in[K]},\quad \vec{\mathsf{r}}^{(W)}, \vec{\mathsf{r}}^{(T)} \xleftarrow{\$} \mathbb{Z}_p^{*K},\quad \mathsf{r}_R, \mathsf{r}_S \xleftarrow{\$} \mathbb{Z}_p^*,$$

$$\mathsf{r}_K^{(W)} \triangleq \sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} - \sum_{k\in[K-1]} \mathsf{r}_k^{(W)},\quad \left(\bar{\tilde{w}}_{j,k} \triangleq \bar{\tilde{b}}_{j,k} \cdot \bar{\tilde{2}}_{j,k}\right)_{j\in[J],k\in[K]}$$

$$\left(t_{j,k} \triangleq \mathsf{r}_{j,k} \cdot (\bar{\tilde{2}}_{j,k} - 2\bar{\tilde{w}}_{j,k})\right)_{j\in[J],k\in[K]}$$

$$\mathcal{P} \Rightarrow \mathcal{V}: \left(W_k \triangleq G^{\sum_{j\in[J]} \bar{\tilde{w}}_{j,k}} \cdot Q^{\mathsf{r}_k^{(W)}},\quad T_k \triangleq \prod_{j\in[J]} H_j^{t_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\right)_{k\in[K]} \tag{12}$$

$$R \triangleq G^{\sum_{j\in[J],k\in[K]} \mathsf{r}_{j,k}} \cdot Q^{\mathsf{r}_R},\quad S \triangleq \prod_{j\in[J]} H_j^{-\sum_{k\in[K]} (\mathsf{r}_{j,k})^2} \cdot Q^{\mathsf{r}_S} \tag{13}$$

$$\mathcal{P} \Leftarrow \mathcal{V}: \vec{e} \xleftarrow{\$} \mathbb{Z}_p^{*K} \tag{14}$$

$$\mathcal{P} \Rightarrow \mathcal{V}: \left(v_{j,k} \triangleq \bar{\tilde{w}}_{j,k} \cdot \epsilon_k + \mathsf{r}_{j,k}\right)_{j\in[J],k\in[K]},\ \eta_1 \triangleq \vec{\mathsf{r}}^{(T)} \cdot \vec{e} + \mathsf{r}_S,\ \eta_2 \triangleq \vec{\mathsf{r}}^{(W)} \cdot \vec{e} + \mathsf{r}_R \tag{15}$$

$$\mathcal{V}: \left(u_{j,k} \triangleq_{j,k} \cdot \epsilon_k - v_{j,k}\right)_{j\in[J],k\in[K]}$$

$$\textsc{Check} \begin{cases} \prod_{j\in[J]} H_j^{\sum_{k\in[K]} v_{j,k} \cdot u_{j,k}} \cdot Q^{\eta_1} \overset{?}{=} \prod_{k\in[K]} T_k^{\epsilon_k} \cdot S \\ G^{\sum_{j\in[J],k\in[K]} v_{j,k}} \cdot Q^{\eta_2} \overset{?}{=} \prod_{k\in[K]} W_k^{\epsilon_k} \cdot R \\ \prod_{t\in[T]} (\mathtt{Cm}(\omega^{(t)}))^{\gamma^t} \overset{?}{=} \prod_{k\in[K]} W_k \end{cases} \tag{16}$$

Although type-1 argument resembles Flashproofs (see Appendix F), it sets different values of J, K, resulting in less group exponentiations at the expense of more field elements in the proof.

3.2 Type-1 Range Argument Protocol

The full protocol of VeRange type-1 range argument is described in Fig. 1, with the steps ①-⑥ labeled in the protocol.

Theorem 3.1. VeRange *type-1 range argument protocol* Π_{ty1} *satisfies perfect completeness, SHVZK and CWE.*

The complete proof can be found in Appendix C.

Remarks: VeRange type-1 range argument differs from Flashproofs (see Appendix F) in the way of checking the satisfiability of $(w_{j,k})$, as to reduce the required group exponentiations. The proof size of includes $2K$ group elements and JK field elements. The verification takes $J + 2K$ group exponentiations. The proving takes $JK + 3K + J$ group exponentiations. To minimize the number of group exponentiations in verification, we set $J \approx K \approx \lceil N^{1/2} \rceil$. Hence, the verification takes around $3N^{1/2}$ group exponentiations and proving takes around $N + 4N^{1/2}$. The proof size includes around $2N^{1/2}$ group elements and N field elements. See Table 1 for a comparison.

3.3 Aggregating Type-1 Range Arguments

Given $(\omega^{(t)})_{t\in[T]}$, the prover commits to $\mathtt{Cm}(\omega^{(t)}) \triangleq G^{\omega^{(t)}} \cdot Q^{\mathsf{r}_{\omega(t)}}$ and aims to prove $\omega^{(t)} \in [0, 2^N - 1]$ for all $t \in [T]$. Rather proving the bit-decomposition of each $\omega^{(t)}$ separately, one can prove the bit-decomposition of $(\omega^{(t)})_{t\in[T]}$ together in a single argument.

Let $\vec{\mathbf{b}}^{(t)}$ be the bit-decomposition of $\omega^{(t)}$. We arrange $(\vec{\mathbf{b}}^{(t)})_{t\in[T]}$ in a $J \times K$ matrix $\left(\bar{\tilde{b}}_{j,k}\right)_{j=1,k=1}^{J\ K}$, as defined by

$$\bar{\tilde{b}}_{j,k} \triangleq \begin{cases} b^{(t)}_{J(k-1)+j-(t-1)N}, & \text{if } (t-1)N < J(k-1)+j \le tN \\ 0, & \text{if } J(k-1)+j > TN \end{cases}$$

Given $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, define a $J \times K$ matrix $\left(\bar{\tilde{2}}_{j,k}\right)_{j=1,k=1}^{J\ K}$ by

$$\bar{\tilde{2}}_{j,k} \triangleq \begin{cases} \gamma^t \cdot 2^{J(k-1)+j-1-(t-1)N}, & \text{if } (t-1)N < J(k-1)+j \le tN \\ 0, & \text{if } J(k-1)+j > TN \end{cases}$$

Also, define $\left(\bar{\tilde{w}}_{j,k} \triangleq \bar{\tilde{b}}_{j,k} \cdot \bar{\tilde{2}}_{j,k}\right)_{j\in[J],k\in[K]}$.

Hence, given a challenge $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, one can check if $\omega^{(t)} \in [0, 2^N - 1]$ for all $t \in [T]$ via random linear combination by checking

$$\prod_{t\in[T]} \left(\mathtt{Cm}(\omega^{(t)})\right)^{\gamma^t} \overset{?}{=} \prod_{k\in[K]} W_k \text{ and } w_{j,k} \overset{?}{\in} \{0, \bar{\tilde{2}}_{j,k}\},\ \forall k \in [K], j \in [J]$$

where $(W_k \triangleq G^{\sum_{j\in[J]} \bar{\tilde{w}}_{j,k}} \cdot Q^{\mathsf{r}_k^{(W)}})_{k\in[K]}$ and the random masks $(\mathsf{r}_k^{(W)})_{k\in[K]}$ are set to satisfy $\sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} = \sum_{k\in[K]} \mathsf{r}_k^{(W)}$.

We provide the full protocol of aggregated type-1 range argument in Fig. 2. The verification takes around $3(TN)^{1/2} + T$ group exponentiations and proving takes $TN + 4(TN)^{1/2}$. The proof size includes around $2(TN)^{1/2}$ group elements and TN field elements.

The aggregated type-1 range argument protocol $\Pi_{\text{a.ty1}}$ can be shown to satisfy perfect completeness, SHVZK and CWE, by extending Theorem 3.1 straightforwardly.

4 VeRange Type-2 Range Argument

In this section, we present the type-2 range argument[3], which is inspired by Bulletproofs++'s approach of reciprocal relation.

4.1 Technical Overview

Instead of bit decomposition, we can also use B-ary digit decomposition, such that if $\omega \in [0, B^{\tilde{N}} - 1]$, then we can express ω by $\omega = \sum_{i\in[\tilde{N}]} d_i \cdot B^{i-1}$, where $d_i \in \{0, 1, ..., B-1\}$. Vector $\vec{\mathbf{d}}$ is called the *B-ary digital decomposition* of ω. For each possible symbol $c \in \{0, 1, ..., B-1\}$ in a B-ary digital decomposition, let m_c be the multiplicity of symbol c appearing in $\vec{\mathbf{d}}$, i.e., $m_c \triangleq \sum_{i\in[\tilde{N}]} \mathbb{1}(d_i = c)$.

It was observed in Bulletproofs++ [18] that $\vec{\mathbf{d}}$ and $\vec{\mathbf{m}}$ should satisfy the following *reciprocal relation*:

$$\sum_{i\in[\tilde{N}]} \frac{1}{\alpha + d_i} = \sum_{c=0}^{B-1} \frac{m_c}{\alpha + c} \tag{17}$$

for any $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. Equivalently, one can check if $\omega \in [0, B^{\tilde{N}} - 1]$ by checking if there exist vectors $(\vec{\mathbf{d}} \in \mathbb{Z}_p^{\tilde{N}}, \vec{\mathbf{m}} \in \mathbb{Z}_p^B)$, such that one can always find $\vec{\mathbf{f}} \in \mathbb{Z}_p^{\tilde{N}}$ for any $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ satisfying the following:

$$\begin{cases} f_i(\alpha + d_i) \stackrel{?}{=} 1, & \forall i \in [\tilde{N}] \\ \sum\limits_{i\in[\tilde{N}]} f_i \stackrel{?}{=} \sum\limits_{c=0}^{B-1} \frac{m_c}{\alpha+c}, & \\ \sum\limits_{i\in[\tilde{N}]} d_i \cdot B^{i-1} \stackrel{?}{=} \omega & \end{cases} \tag{18}$$

We next outline the basic idea of an efficient zero-knowledge proof protocol that checks Eqn. (18) with respect to a commitment of ω, i.e., $\mathtt{Cm}(\omega) \triangleq G^{\omega} \cdot Q^{\mathsf{r}_\omega}$, without revealing $\vec{\mathbf{d}}$ or $\vec{\mathbf{m}}$.

We arrange $\vec{\mathbf{d}}$ in a $\tilde{J} \times \tilde{K}$ matrix $\left(\hat{d}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$, as defined by

$$\hat{d}_{j,k} \triangleq \begin{cases} d_{\tilde{J}(k-1)+j}, & \text{if } \tilde{J}(k-1)+j \le \tilde{N} \\ 0, & \text{if } \tilde{J}(k-1)+j > \tilde{N} \end{cases} \tag{19}$$

Also, define a $\tilde{J} \times \tilde{K}$ matrix $\left(\hat{B}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$ by

$$\hat{B}_{j,k} \triangleq \begin{cases} B^{\tilde{J}(k-1)+j-1}, & \text{if } \tilde{J}(k-1)+j \le \tilde{N} \\ 0, & \text{if } \tilde{J}(k-1)+j > \tilde{N} \end{cases} \tag{20}$$

Let $\mathcal{B} \triangleq \left\{(j,k) \in [\tilde{J}] \times [\tilde{K}] : \hat{B}_{i,j} \neq 0\right\}$.

Then, define $\left(\tilde{w}_{j,k} \triangleq \hat{d}_{j,k} \cdot \hat{B}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$, namely,

$$\begin{pmatrix} \tilde{w}_{1,1} & \dots & \tilde{w}_{1,\tilde{K}} \\ \vdots & \ddots & \vdots \\ \tilde{w}_{\tilde{J},1} & \dots & \tilde{w}_{\tilde{J},\tilde{K}} \end{pmatrix} \triangleq \begin{pmatrix} d_1 \cdot B^0 & d_{\tilde{J}} \cdot B^{\tilde{J}+1} & \dots & d_{\tilde{J}(\tilde{K}-1)+1} \cdot B^{\tilde{J}(\tilde{K}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{\tilde{J}-\tilde{\eta}} \cdot B^{\tilde{J}-\tilde{\eta}-1} & d_{2\tilde{J}-\tilde{\eta}} \cdot B^{2\tilde{J}-\tilde{\eta}-1} & \dots & d_{\tilde{N}} \cdot B^{\tilde{N}-1} \\ d_{\tilde{J}-\tilde{\eta}+1} \cdot B^{\tilde{J}-\tilde{\eta}} & d_{2\tilde{J}-\tilde{\eta}+1} \cdot B^{2\tilde{J}-\tilde{\eta}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d_{\tilde{J}} \cdot B^{\tilde{J}-1} & d_{2\tilde{J}} \cdot B^{2\tilde{J}-1} & \dots & 0 \end{pmatrix}$$

where $\tilde{\eta} = \tilde{J}\tilde{K} \bmod \tilde{N}$, such that $\tilde{\eta} < \tilde{J}$.

Given $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$, let $f_{j,k} \triangleq \frac{1}{\alpha + \hat{d}_{j,k}}$. We then can re-express the first equation of Eqn. (18) by the following to relate $f_{j,k}$ and $\tilde{w}_{j,k}$:

$$\begin{aligned} & 1 \stackrel{?}{=} B^{-i+1} \cdot f_i \cdot (\alpha \cdot B^{i-1} + d_i \cdot B^{i-1}),\ \forall i \in [\tilde{N}] \\ \Rightarrow\ & 1 \stackrel{?}{=} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot (\alpha \cdot \hat{B}_{j,k} + \hat{d}_{j,k} \cdot \hat{B}_{j,k}),\ \forall (j,k) \in \mathcal{B} \\ & = \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \end{aligned} \tag{21}$$

Hence, Eqn. (18) becomes

$$\begin{cases} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \stackrel{?}{=} 1, & \forall (j,k) \in \mathcal{B} \\ \sum\limits_{(j,k)\in\mathcal{B}} f_{i,k} \stackrel{?}{=} \sum\limits_{c=0}^{B-1} \frac{m_c}{\alpha+c}, & \\ \sum\limits_{j\in[\tilde{J}],k\in[\tilde{J}]} \tilde{w}_{j,k} \stackrel{?}{=} \omega & \end{cases} \tag{22}$$

In this range argument, the prover first commits $(\sum_{j\in[J]} \tilde{w}_{j,k})_{k\in[K]}$ as $(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \tilde{w}_{j,k}} \cdot Q^{\mathsf{r}_k^{(\Omega)}})_{k\in[\tilde{K}]}$, where the random masks $(\mathsf{r}_k^{(\Omega)})_{k\in[\tilde{K}]}$ are set to satisfy $\mathsf{r}_\omega = \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(\Omega)}$. Then, the verifier can check if $\omega \in [0, B^{\tilde{N}} - 1]$ by checking

$$\mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \ \text{ and } \ \tilde{w}_{j,k} \stackrel{?}{\in} \{0, \hat{B}_{j,k}, ..., (B-1)\cdot\hat{B}_{j,k}\}, \forall (j,k) \in \mathcal{B}$$

Next, we proceed to check the satisfiability of $(\tilde{w}_{j,k} \in \{0, \hat{B}_{j,k}, ..., (B-1) \cdot \hat{B}_{j,k}\})_{(j,k)\in\mathcal{B}}$ by the below zero-knowledge protocol:

1. In addition to $(\Omega_k)_{k\in[\tilde{K}]}$, the prover also commits $(M_c \triangleq G^{m_c} \cdot Q^{\mathsf{r}_c^{(M)}})_{c=0}^{B-1}$ and $\left((V_k)_{k\in[\tilde{K}]}, \tilde{R}, \tilde{S}\right)$, which will be defined in the following.
2. The verifier then sends a random number $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ to the prover.
3. The prover next computes $f_{j,k} \triangleq \frac{1}{\alpha+\hat{d}_{j,k}}$ and commits $(\tilde{T}_k)_{k\in[\tilde{K}]}$, which will be defined in the following.
4. The verifier then sends a random challenge vector $\vec{\epsilon} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}$ to the prover.
5. The prover replies with the following:

$$\left(v_{j,k} \triangleq (\alpha\cdot\hat{B}_{j,k}+\tilde{w}_{j,k})\cdot\epsilon_k+\mathsf{r}_{j,k}^{(v)},\ \mu_{j,k} \triangleq \hat{B}_{j,k}^{-1}\cdot f_{j,k}\cdot\epsilon_k+\mathsf{r}_{j,k}^{(\mu)}\right)_{(j,k)\in\mathcal{B}}$$

where $\mathsf{r}_{j,k}^{(\mu)}, \mathsf{r}_{j,k}^{(v)} \xleftarrow{\$} \mathbb{Z}_p$ are random mask numbers.

6. Next, the satisfiability of the first equation of Eqn. (22) can be checked by the following:

$$\begin{aligned} \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} v_{j,k}\cdot\mu_{j,k} \stackrel{?}{=} & \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \underbrace{\hat{B}_{j,k}^{-1}\cdot f_{j,k}\cdot(\alpha\cdot\hat{B}_{j,k}+\tilde{w}_{j,k})}_{=1 \text{ if Eqn. (21) is satisfied}}\cdot\epsilon_k^2 \\ & + \sum_{k\in[\tilde{K}]} \left(\hat{B}_{j,k}^{-1}\cdot f_{j,k}\cdot\mathsf{r}_{j,k}^{(v)} + (\alpha\cdot\hat{B}_{j,k}+\tilde{w}_{j,k})\cdot\mathsf{r}_{j,k}^{(\mu)}\right)\cdot\epsilon_k \\ & + \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \mathsf{r}_{j,k}^{(\mu)}\cdot\mathsf{r}_{j,k}^{(v)} \end{aligned}$$

By the DLR assumption, one can equivalently check the following equation:

$$\begin{aligned} & \textstyle\prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} v_{j,k}\cdot\mu_{j,k}+\sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}}\epsilon_k^2} \\ & \stackrel{?}{=} \left(\textstyle\prod_{j\in[\tilde{J}]} H_j\right)^{\sum_{k\in[\tilde{K}]}\epsilon_k^2} \cdot \textstyle\prod_{k\in[\tilde{K}]} \tilde{T}_k^{\epsilon_k} \cdot \tilde{S}, \end{aligned}$$

[3] We also present the type-2B range argument in Appendix G.1.

where

$$\tau_{j,k} \triangleq \begin{cases} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot \mathsf{r}_{j,k}^{(\nu)} + (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \cdot \mathsf{r}_{j,k}^{(\mu)}, & \text{if } (j,k) \in \mathcal{B} \\ 0, & \text{if } (j,k) \notin \mathcal{B} \end{cases},$$

$\tilde{T}_k \triangleq \prod_{j\in[\tilde{J}]} H_j^{\tau_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}$, $\tilde{S} \triangleq \prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \mathsf{r}_{j,k}^{(\mu)}\cdot\mathsf{r}_{j,k}^{(\nu)}} \cdot Q^{\mathsf{r}_S}$

$(\tilde{T}_k)_{k\in[\tilde{K}]}$ and $\tilde{S}$ should be committed by the prover at ①,③ before knowing $\vec{\epsilon}$ and the prover provides $\tilde{\eta}_1 \triangleq \vec{\mathsf{r}}^{(T)} \cdot \vec{\epsilon} + \mathsf{r}_S$.

⑦ The satisfiability of the second equation of Eqn. (22) can be checked by the following:

$$\sum_{(j,k)\in\mathcal{B}} \hat{B}_{j,k} \cdot \mu_{j,k} \cdot \epsilon_k^{-1} \stackrel{?}{=} \sum_{c=0}^{B-1} \frac{m_c}{\alpha + c} + \sum_{(j,k)\in\mathcal{B}} \hat{B}_{j,k} \cdot \mathsf{r}_{j,k}^{(\mu)} \cdot \epsilon_k^{-1} \tag{23}$$

By the DLR assumption, one can equivalently check the following equation:

$$G^{\sum_{(j,k)\in\mathcal{B}} \hat{B}_{j,k}\cdot\mu_{j,k}\cdot\epsilon_k^{-1}} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} \prod_{c=0}^{B-1} M_c^{\frac{1}{\alpha+c}} \cdot \prod_{k\in[\tilde{K}]} V_k^{\epsilon_k^{-1}} \tag{24}$$

where $V_k \triangleq G^{\sum_{(j,k)\in\mathcal{B}} \hat{B}_{j,k}\cdot\mathsf{r}_{j,k}^{(\mu)}} \cdot Q^{\mathsf{r}_k^{(V)}}$ should be committed by the prover at ① before knowing α and the prover provides $\tilde{\eta}_2 \triangleq \sum_{c=0}^{B-1} \frac{\mathsf{r}_c^{(M)}}{\alpha+c} + \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(V)} \cdot \epsilon_k^{-1}$.

4.2 Type-2 Range Argument Protocol

The full protocol of VeRange type-2 range argument is described in Fig. 3, with the steps ①-⑦ labeled in the protocol.

Theorem 4.1. VeRange *type-2 range argument protocol* Π_{ty2} *satisfies perfect completeness, SHVZK and CWE.*

The complete proof can be found in Appendix D.

Remarks: The proof size of VeRange type-2 range argument includes $3\tilde{K} + B$ group elements and $2\tilde{J}\tilde{K}$ field elements. The verification takes $\tilde{J} + 3\tilde{K} + B$ group exponentiations. The proving takes $\tilde{J}\tilde{K} + 5\tilde{K} + \tilde{J} + 2B$ group exponentiations. To minimize the number of group exponentiations in verification, we set $B \approx (\frac{N}{\log N})^{1/2}$. Since $B^{\tilde{N}} = 2^N$, we obtain $\tilde{N} = \frac{N}{\log B} \approx \frac{2N}{\log N}$ and set $\tilde{J} \approx \tilde{K} \approx \left\lceil (\frac{2N}{\log N})^{1/2} \right\rceil$. Hence, the verification takes around $(4\sqrt{2}+1)(\frac{N}{\log N})^{1/2}$ group exponentiations and proving takes around $2\frac{N}{\log N} + (6\sqrt{2}+2)(\frac{N}{\log N})^{1/2}$. The proof size includes around $(3\sqrt{2}+1)(\frac{N}{\log N})^{1/2}$ group elements and $4\frac{N}{\log N}$ field elements. See Table 1 for a comparison.

4.3 Aggregating Type-2 Range Arguments

Multiple type-2 range arguments can be aggregated in a similar manner as type-1 in Sec. 3.3. Given $(\omega^{(t)})_{t\in[T]}$, the prover commits to $\mathtt{Cm}(\omega^{(t)}) \triangleq G^{\omega^{(t)}} \cdot Q^{\mathsf{r}_{\omega(t)}}$ and aims to prove $\omega^{(t)} \in [0, B^{\tilde{N}} - 1]$ for all $t \in [T]$. Let $\vec{\mathbf{d}}^{(t)}$ be the B-ary digit decomposition of $\omega^{(t)}$. We arrange $(\vec{\mathbf{d}}^{(t)})_{t\in[T]}$ in a $\tilde{J} \times \tilde{K}$ matrix $\left(\bar{\tilde{d}}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}$, defined by

$$\bar{\tilde{d}}_{j,k} \triangleq \begin{cases} d^{(t)}_{\tilde{J}(k-1)+j-(t-1)\tilde{N}}, & \text{if } (t-1)\tilde{N} < \tilde{J}(k-1) + j \le t\tilde{N} \\ 0, & \text{if } \tilde{J}(k-1) + j > T\tilde{N} \end{cases}$$

Figure 3: VeRange **type-2 range argument protocol**

$\Pi_{\text{ty2}}\left[\mathtt{Cm}(\omega) \in \mathbb{G};\ \omega \in \mathbb{Z}_p, \mathsf{r}_\omega \in \mathbb{Z}_p^*\right]$

Setup : $H_{\tilde{J}} \triangleq \prod_{j\in[\tilde{J}]} H_j$

$\mathcal{P} : \vec{\mathbf{d}} \in (\{0, ..., B-1\})^{\tilde{N}}$ is the B-ary digit decompo. of ω such that $\omega = \sum_{i\in[N]} d_i \cdot B^{i-1}$

$\left(\mathsf{r}_{j,k}^{(\mu)}, \mathsf{r}_{j,k}^{(\nu)} \stackrel{\$}{\leftarrow} \mathbb{Z}_p\right)_{j\in[\tilde{J}],k\in[\tilde{K}]}$, $\vec{\mathsf{r}}^{(\Omega)}, \vec{\mathsf{r}}^{(T)}, \vec{\mathsf{r}}^{(V)} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{*\tilde{K}}$, $\vec{\mathsf{r}}^{(M)} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{*B}$

$\mathsf{r}_R, \mathsf{r}_S \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, $\mathsf{r}_{\tilde{K}}^{(\Omega)} \triangleq \mathsf{r}_\omega - \sum_{k\in[\tilde{K}-1]} \mathsf{r}_k^{(\Omega)}$

$\left(\tilde{w}_{j,k} \triangleq \dot{d}_{j,k} \cdot \hat{B}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}$, $\left(m_c \triangleq \sum_{i\in[\tilde{N}]} \mathbb{1}(d_i = c)\right)_{c=0}^{B-1}$

① $\mathcal{P} \Rightarrow \mathcal{V} : \left(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \tilde{w}_{j,k}} \cdot Q^{\mathsf{r}_k^{(\Omega)}}\right)_{k\in[\tilde{K}]}, \left(M_c \triangleq G^{m_c} \cdot Q^{\mathsf{r}_c^{(M)}}\right)_{c=0}^{B-1}$ (25)

$\left(V_k \triangleq G^{\sum_{(j,k)\in\mathcal{B}} \hat{B}_{j,k}\cdot\mathsf{r}_{j,k}^{(\mu)}} \cdot Q^{\mathsf{r}_k^{(V)}}\right)_{k\in[\tilde{K}]}$ (26)

$\tilde{R} \triangleq G^{\sum_{(j,k)\in\mathcal{B}} \mathsf{r}_{j,k}^{(\nu)}} \cdot Q^{\mathsf{r}_R}$, $\tilde{S} \triangleq \prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \mathsf{r}_{j,k}^{(\mu)}\cdot\mathsf{r}_{j,k}^{(\nu)}} \cdot Q^{\mathsf{r}_S}$ (27)

② $\mathcal{P} \Leftarrow \mathcal{V} : \alpha \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ (28)

③ $\mathcal{P} : \left(f_{j,k} \triangleq \frac{1}{\alpha + \dot{d}_{j,k}}\right)_{(j,k)\in\mathcal{B}}$

$\tau_{j,k} \triangleq \begin{cases} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot \mathsf{r}_{j,k}^{(\nu)} + (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \cdot \mathsf{r}_{j,k}^{(\mu)}, & \text{if } (j,k) \in \mathcal{B} \\ 0, & \text{if } (j,k) \notin \mathcal{B} \end{cases}$

$\mathcal{P} \Rightarrow \mathcal{V} : \left(\tilde{T}_k \triangleq \prod_{j\in[\tilde{J}]} H_j^{\tau_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\right)_{k\in[\tilde{K}]}$

④ $\mathcal{P} \Leftarrow \mathcal{V} : \vec{\epsilon} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{*\tilde{K}}$ (29)

⑤ $\mathcal{P} \Rightarrow \mathcal{V} : \left(\nu_{j,k} \triangleq (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \cdot \epsilon_k + \mathsf{r}_{j,k}^{(\nu)},\ \mu_{j,k} \triangleq \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot \epsilon_k + \mathsf{r}_{j,k}^{(\mu)}\right)_{(j,k)\in\mathcal{B}}$ (30)

$\tilde{\eta}_1 \triangleq \vec{\mathsf{r}}^{(T)} \cdot \vec{\epsilon} + \mathsf{r}_S,\ \tilde{\eta}_2 \triangleq \sum_{c=0}^{B-1} \frac{\mathsf{r}_c^{(M)}}{\alpha + c} + \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(V)} \cdot \epsilon_k^{-1},\ \tilde{\eta}_3 \triangleq \vec{\mathsf{r}}^{(\Omega)} \cdot \vec{\epsilon} + \mathsf{r}_R$ (31)

$\mathcal{V} : \text{Check}$

⑥ $\prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \nu_{j,k}\cdot\mu_{j,k} + \sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}} \epsilon_k^2} \cdot Q^{\tilde{\eta}_1} \stackrel{?}{=} H_{\tilde{J}}^{\sum_{k\in[\tilde{K}]} \epsilon_k^2} \cdot \prod_{k\in[\tilde{K}]} \tilde{T}_k^{\epsilon_k} \cdot \tilde{S}$

⑦ $G^{\sum_{(j,k)\in\mathcal{B}} \hat{B}_{j,k}\cdot\mu_{j,k}\cdot\epsilon_k^{-1}} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} \prod_{c=0}^{B-1} M_c^{\frac{1}{\alpha+c}} \cdot \prod_{k\in[\tilde{K}]} V_k^{\epsilon_k^{-1}}$

$G^{\sum_{(j,k)\in\mathcal{B}} \nu_{j,k} - \alpha\cdot\hat{B}_{j,k}\cdot\epsilon_k} \cdot Q^{\tilde{\eta}_3} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k^{\epsilon_k} \cdot \tilde{R}$

$\mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k$ (32)

Given $\gamma \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, define a $\tilde{J} \times \tilde{K}$ matrix $\left(\bar{\tilde{B}}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}$ by

$$\bar{\tilde{B}}_{j,k} \triangleq \begin{cases} \gamma^t \cdot B^{\tilde{J}(k-1)+j-1-(t-1)\tilde{N}}, & \text{if } (t-1)\tilde{N} < \tilde{J}(k-1) + j \le t\tilde{N} \\ 0, & \text{if } \tilde{J}(k-1) + j > T\tilde{N} \end{cases}$$

Also, define $\left(\bar{\tilde{w}}_{j,k} \triangleq \bar{\tilde{d}}_{j,k} \cdot \bar{\tilde{B}}_{j,k}\right)_{j\in[\tilde{J}],k\in[\tilde{K}]}$.

Given a challenge $\gamma \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, one can check if $\omega^{(t)} \in [0, B^{\tilde{N}} - 1]$ for all $t \in [T]$ via random linear combination by checking

$$\prod_{t\in[T]} \left(\mathtt{Cm}(\omega^{(t)})\right)^{\gamma^t} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \text{ and } \bar{\tilde{w}}_{j,k} \stackrel{?}{\in} \{0, \bar{\tilde{B}}_{j,k}, ..., (B\text{-}1)\cdot\bar{\tilde{B}}_{j,k}\}, \forall (j,k) \in \mathcal{B}$$

where $(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \vec{w}_{j,k}} \cdot Q^{\mathsf{r}_k^{(\Omega)}})_{k\in[\tilde{K}]}$ and the random masks $(\mathsf{r}_k^{(\Omega)})_{k\in[\tilde{K}]}$ are set to satisfy $\sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} = \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(\Omega)}$.

Extending the type-2 range argument protocol, we can construct the full protocol of aggregated type-2 range argument in Fig. 11 in Appendix G.2. The verification takes $O(\sqrt{\frac{TN}{\log(TN)}}) + T$ group exponentiations and proving takes $O(\frac{TN}{\log(TN)})$. The proof size includes $O(\sqrt{\frac{TN}{\log(TN)}})$ group elements and field elements.

The aggregated type-2 range argument protocol can be shown to satisfy perfect completeness, SHVZK and CWE, by extending Theorem 4.1 straightforwardly.

5 VeRange Type-3 Range Argument

In this section, we present the type-3 range argument, which is based on the idea of efficient batch verification of polynomial evaluation. Although our approach is based on BG18 [5], our approach differs from BG18, as we especially optimize the batch verification to reduce group exponentiations in verification.

5.1 Technical Overview

By B-ary digit decomposition, one can check if $\omega \in [0, B^{\tilde{N}} - 1]$ by checking if there exists a vector $\vec{\mathbf{d}} \in \mathbb{Z}_p^{\tilde{N}}$, such that

$$\begin{cases} d_i \cdot (d_i - 1) \cdots (d_i - B + 1) \stackrel{?}{=} 0, & \text{for all } i \in [\tilde{N}] \\ \sum\limits_{i\in\tilde{N}} d_i \cdot B^{i-1} \stackrel{?}{=} \omega \end{cases} \tag{33}$$

Checking Eqn. (33) can be thought of as performing polynomial evaluation. We will utilize efficient batch verification of polynomial evaluation through a polynomial commitment to check Eqn. (33). Next, we introduce the basics of a polynomial commitment scheme and its application for batch verification of polynomial evaluation.

Polynomial Commitment: A polynomial commitment scheme allows a prover to commit to a polynomial (as a secret) in advance and to open the evaluation at a specific point subsequently with a proof to show that the evaluated polynomial is identical to the one committed. A generic polynomial commitment scheme consists of four methods (`Setup`, `PolyCm`, `PolyEv`, `PolyVf`).

Given polynomial $\mathsf{F}[X] \triangleq \sum_{i=0}^{D} h_i X^i$, we randomly generate $\mathsf{r}_1, ..., \mathsf{r}_V \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ as random masks. Define a $(U+1)\times(V+1)$ matrix $(\hat{h}_{u,v})_{u=0,v=0}^{U\ V}$ as follows:

$$\begin{pmatrix} \hat{h}_{0,0} & \cdots & \hat{h}_{0,V} \\ \vdots & \ddots & \vdots \\ \hat{h}_{U,0} & \cdots & \hat{h}_{U,V} \end{pmatrix} \triangleq \begin{pmatrix} h_0 & \mathsf{r}_1 & \cdots & \mathsf{r}_{V-1} & \mathsf{r}_V \\ h_1 & h_{\xi+1} & \cdots & h_{U(V-2)+\xi+1} & h_{U(V-1)+\xi+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_\xi - \mathsf{r}_1 & h_{2\xi} & \cdots & h_{U(V-2)+\xi} & h_{U(V-1)+2\xi} \\ 0 & h_{2\xi+1} & \cdots & h_{U(V-2)+\xi+1} & h_{U(V-1)+2\xi+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & h_{U+\xi} - \mathsf{r}_2 & \cdots & h_{U(V-1)+\xi} - \mathsf{r}_V & h_D \end{pmatrix}$$

where $\xi = (U+1)(V+1) \bmod (D+1)$. Note that we can re-express $\mathsf{F}[X]$ as

$$\mathsf{F}[X] = \sum_{u=0}^{U} \hat{h}_{u,0} X^u + \sum_{v=1}^{V} \Big(\sum_{u=0}^{U} \hat{h}_{u,v} X^u \Big) X^{(v-1)U+\xi}$$

In this range argument, we will utilize the BCCGP polynomial commitment scheme [4] (as described in Fig. 4), which satisfies computational binding and perfect hiding. BCCGP polynomial commitment scheme takes $(U+V+2)$ group exponentiations to verify an evaluation of a committed polynomial with degree D, with a commitment size of $U+1$ group elements and a proof size of $V+1$ field elements. BCCGP commitment generation takes $(U+1)(V+1)$ group exponentiations. To minimize group exponentiations in verification, we set $U \approx V \approx \lceil\sqrt{D}\rceil$. Hence, verification of BCCGP polynomial commitment takes around $2\sqrt{D}$ group exponentiations and proving takes around D group exponentiations. The commitment size includes around $\sqrt{D}$ group elements and the proof size includes around $\sqrt{D}$ field elements.

Figure 4: BCCGP polynomial commitment scheme [4]

- $\mathsf{Setup}_{\mathsf{BCCGP}}$: The public parameters for BCCGP scheme are $(V+1)$ random generators from $\mathbb{G}$:
$$\mathsf{pp} \leftarrow (G_v \stackrel{\$}{\leftarrow} \mathbb{G})_{v=0}^{V}$$
- $\mathsf{PolyCm}_{\mathsf{BCCGP}}$: Given a polynomial $\mathsf{F}[X] \triangleq \sum_{i=0}^{D} h_i X^i$, the commitment of F has $(U+1)$ group elements:
$$\mathtt{Cm}_{\mathsf{F}} \leftarrow \Big(H_u \triangleq \prod_{v=0}^{V} G_v^{\hat{h}_{u,v}} \Big)_{u=0}^{U} \in \mathbb{G}^{U+1}$$
- $\mathsf{PolyEv}_{\mathsf{BCCGP}}$: The proof π_{F} to the evaluation $y = \mathsf{F}[x]$ for commitment $\mathtt{Cm}_{\mathsf{F}}$ has $(V+1)$ field elements:
$$\pi_{\mathsf{F}} \leftarrow \Big(\mathsf{f}_v \triangleq \sum_{u=0}^{U} \hat{h}_{u,v} \cdot x^u \Big)_{v=0}^{V} \in \mathbb{Z}_p^{V+1}$$
- $\mathsf{PolyVf}_{\mathsf{BCCGP}}$: To verify $(\mathsf{pp}, \mathtt{Cm}_{\mathsf{F}}, x, y, \pi_{\mathsf{F}})$, the verifier checks the following equations:
$$\begin{cases} \prod\limits_{v=0}^{V} G_v^{\mathsf{f}_v} \stackrel{?}{=} \prod\limits_{u=0}^{U} H_u^{x^u} \\ \mathsf{f}_0 + \sum\limits_{v=1}^{V} \mathsf{f}_v \cdot x^{(v-1)U+\xi} \stackrel{?}{=} y \end{cases} \tag{34}$$
$\mathsf{PolyVf}_{\mathsf{BCCGP}}$ returns 1, if the above equations are equal, or 0 otherwise.

Lagrange Polynomials: We will also utilize Lagrange polynomials as a way to enable batch verification of polynomial evaluation. Let $\{z_i \in \mathbb{Z}_p\}_{i=0}^{m}$ be a set of $m+1$ distinct values in $\mathbb{Z}_p$. We define a *Lagrange basis polynomial* by

$$\mathsf{L}_i[X] \triangleq \prod_{j\in\{0,\ldots,m\}\setminus\{i\}} \frac{X - z_j}{z_i - z_j}$$

where $i \in [m]$ is an index of Lagrange basis polynomials. Note that $\mathsf{L}_i[z_j] = 0$, if $i \neq j$, and $\mathsf{L}_i[z_i] = 1$. We also define

$$\mathsf{L}_0[X] \triangleq \prod_{j\in[m]} (X - z_j)$$

Note that $\{z_i\}_{i\in[m]}$ are the roots of $\mathsf{L}_0[X]$. As a result, there is a way to efficiently aggregate the evaluation of multiple input values for a polynomial. Suppose $\{a^{(i)} \in \mathbb{Z}_p\}_{i\in[m]}$ are the roots of polynomial $\mathsf{F}[X]$, i.e., $\mathsf{F}[a^{(i)}] = 0$ for all $i \in [m]$. We encode $\{a^{(i)}\}_{i\in[m]}$ into a polynomial as follows:

$$\bar{a}[X] \triangleq \sum_{i\in[m]} a^{(i)} \cdot \mathsf{L}_i[X]$$

Note that $\{z_i\}_{i\in[m]}$ are also the roots of $\mathsf{F}\big[\bar{a}[X]\big]$, i.e., $\mathsf{F}\big[\bar{a}[z_i]\big] = \mathsf{F}[a^{(i)}] = 0$ for all $i \in [m]$. Then, this implies that

$$\mathsf{F}\big[\bar{a}[X]\big] \bmod \mathsf{L}_0[X] = 0$$

Hence, rather than checking $\mathsf{F}[a^{(i)}] \stackrel{?}{=} 0$ for all $i \in [m]$, one can probabilistically check $\mathsf{F}\big[\bar{a}[x]\big] \bmod \mathsf{L}_0[x] \stackrel{?}{=} 0$, given a random challenge $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$.

Efficient Batch Verification of Polynomial Evaluation: By the factor theorem of polynomials, given $\{(a^{(i)}, c^{(i)})\}_{i\in[m]}$, if

$$\Big(\sum_{i\in[m]} c^{(i)}\cdot \mathsf{L}_i[X] - \mathsf{F}\big[\bar{a}[X]\big]\Big) \bmod \mathsf{L}_0[X] = 0 \tag{35}$$

then $\mathsf{F}[a^{(i)}] = c^{(i)}$ for all $i \in [m]$. Note that Eqn. (35) is equivalent to $\sum_{i\in[m]} c^{(i)}\cdot \mathsf{L}_i[X] - \mathsf{F}\big[\bar{a}[X]\big]$ being divisible by $\mathsf{L}_0[X]$. Define

$$\mathsf{P}[X] \triangleq \frac{\sum_{i\in[m]} c^{(i)}\cdot \mathsf{L}_i[X] - \mathsf{F}\big[\bar{a}[X]\big]}{\mathsf{L}_0[X]}$$

We now apply Lagrange polynomials to enable efficient batch verification of polynomial evaluation in the following scenarios:

1 ***Open Input Values:*** We consider the evaluation of multiple open input values $\{a^{(i)}\}_{i\in[m]}$ on a known polynomial $\mathsf{F}[X]$ with open output values $\{c^{(i)}\}_{i\in[m]}$. To check $\mathsf{F}[a^{(i)}] \stackrel{?}{=} c^{(i)}$ for all $i \in [m]$, the prover should first commit $\mathsf{P}[X]$ by $\mathtt{Cm}_\mathsf{P} \triangleq \mathtt{PolyCm}[\mathsf{P}]$. Then the verifier issues a random challenge $x \xleftarrow{\$} \mathbb{Z}_p$ and asks the prover to evaluate $y_\mathsf{P} = \mathsf{P}[x]$ and produce a proof $\pi_\mathsf{P} \triangleq \mathtt{PolyEv}[\mathsf{P}, x]$. The prover also returns $\bar{a} \triangleq \bar{a}[x]$. The verifier can verify $(\bar{a}, \{c^{(i)}\}_{i\in[m]})$ by checking the following

$$\begin{cases} 1 \stackrel{?}{=} \mathtt{PolyVf}[\mathtt{Cm}_\mathsf{P}, x, y_\mathsf{P}, \pi_\mathsf{P}] \\ y_\mathsf{P}\cdot \mathsf{L}_0[x] \stackrel{?}{=} \Big(\sum_{i\in[m]} c^{(i)}\cdot \mathsf{L}_i[x] - \mathsf{F}[\bar{a}]\Big) \end{cases}$$

2 ***Secret Input Values:*** We consider the evaluation of multiple *secret* input values $\{a^{(i)}\}_{i\in[m]}$ on a known polynomial $\mathsf{F}[X]$ with open output values $\{c^{(i)}\}_{i\in[m]}$. To mask $\{a^{(i)}\}_{i\in[m]}$, the prover sets a new function for $\bar{a}[X]$:

$$\bar{a}[X] \triangleq \sum_{i\in[m]} a^{(i)}\cdot \mathsf{L}_i[X] + \mathsf{r}_\mathsf{a}\cdot \mathsf{L}_0[X]$$

where $\mathsf{r}_\mathsf{a} \xleftarrow{\$} \mathbb{Z}_p$ is a random mask. The prover commits $\{a^{(i)}\}_{i\in[m]}$ and r_a by $(A_i \triangleq G^{a^{(i)}}\cdot Q^{\mathsf{r}_i})_{i\in[m]}$ and $R \triangleq G^{\mathsf{r}_\mathsf{a}}\cdot Q^{\mathsf{r}}$. Then the verifier issues a random challenge $x \xleftarrow{\$} \mathbb{Z}_p$. The prover returns $y_\mathsf{P} = \mathsf{P}[x]$, $\pi_\mathsf{P} \triangleq \mathtt{PolyEv}[\mathsf{P}, x]$ and $\bar{a} \triangleq \bar{a}[x]$, as in the above scenario. The verifier can verify $\{A_i, c^{(i)}\}_{i\in[m]}$ by checking the following

$$\begin{cases} 1 \stackrel{?}{=} \mathtt{PolyVf}[\mathtt{Cm}_\mathsf{P}, x, y_\mathsf{P}, \pi_\mathsf{P}] \\ G^{\bar{a}}\cdot Q^{\eta} \stackrel{?}{=} \prod_{i\in[m]} A_i^{\mathsf{L}_i[x]}\cdot R^{\mathsf{L}_0[x]} \\ y_\mathsf{P}\cdot \mathsf{L}_0[x] \stackrel{?}{=} \Big(\sum_{i\in[m]} c^{(i)}\cdot \mathsf{L}_i[x] - \mathsf{F}[\bar{a}]\Big) \end{cases}$$

where $\eta \triangleq \sum_{i\in[m]} \mathsf{r}_i\cdot \mathsf{L}_i[x] + \mathsf{r}\cdot \mathsf{L}_0[x]$ is provided by the prover.

3 ***Secret Input and Output Values:*** We consider the evaluation of multiple *secret* input values $\{a^{(i)}\}_{i\in[m]}$ on known $\mathsf{F}[X]$ with *secret* output values $\{c^{(i)}\}_{i\in[m]}$. In addition to committing $\{a^{(i)}\}_{i\in[m]}$ and r_a by $(A_i)_{i\in[m]}$ and R, the prover also commits $\{c^{(i)}\}_{i\in[m]}$ by $(C_i = G^{c^{(i)}}\cdot Q^{\mathsf{r}_i^{(c)}})_{i\in[m]}$. In addition to masking $\{a^{(i)}\}_{i\in[m]}$, the prover masks $\mathsf{P}[X]$ by

$$\mathsf{P}[X] \triangleq \mathsf{s} + \frac{\sum_{i\in[m]} c^{(i)}\cdot \mathsf{L}_i[X] - \mathsf{F}\big[\bar{a}[X]\big]}{\mathsf{L}_0[X]}$$

where $\mathsf{s} \xleftarrow{\$} \mathbb{Z}_p$ is a random mask. The prover returns $y_\mathsf{P} = \mathsf{P}[x]$, $\pi_\mathsf{P} \triangleq \mathtt{PolyEv}[\mathsf{P}, x]$ and $\bar{a} \triangleq \bar{a}[x]$, as in the above scenarios. The verifier can verify $\{A_i, C_i\}_{i\in[m]}$ by checking the following

$$\begin{cases} 1 \stackrel{?}{=} \mathtt{PolyVf}[\mathtt{Cm}_\mathsf{P}, x, y_\mathsf{P}, \pi_\mathsf{P}] \\ G^{\bar{a}}\cdot Q^{\eta_1} \stackrel{?}{=} \prod_{i\in[m]} A_i^{\mathsf{L}_i[x]}\cdot R_1^{\mathsf{L}_0[x]} \\ G^{y_\mathsf{P}\cdot \mathsf{L}_0[x] + \mathsf{F}[\bar{a}]}\cdot Q^{\eta_2} \stackrel{?}{=} \prod_{i\in[m]} (C_i)^{\mathsf{L}_i[x]}\cdot R_2^{\mathsf{L}_0[x]} \end{cases} \tag{36}$$

where $R_2 \triangleq G^{\mathsf{s}}\cdot Q^{\mathsf{r}_2}$ should be committed by the prover before knowing challenge x and $\eta_2 \triangleq \sum_{i\in[m]} \mathsf{r}_i^{(c)}\cdot \mathsf{L}_k[x] + \mathsf{r}_2\cdot \mathsf{L}_0[x]$ is provided by the prover.

Application to Range Arguments: We next apply efficient batch verification of polynomial evaluation to design an efficient range argument and optimize the batch verification to reduce group exponentiations in verification.

Let $\vec{\mathbf{d}} \triangleq (d_1, \ldots d_{\tilde{N}})$ be the B-ary digit decomposition of ω. Suppose $\tilde{\xi} = \tilde{J}\tilde{K} \bmod \tilde{N}$. As in type-2 argument, we arrange $\vec{\mathbf{d}}$ and $\vec{\mathbf{B}}$ in $\tilde{J}\times\tilde{K}$ matrices $(\hat{d}_{j,k})_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}$ and $(\hat{B}_{j,k})_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}$, respectively (see Eqns (19)-(20)). We define the following functions $\mathsf{B}_{j,k}[\cdot], \mathsf{S}_k[\cdot]$:

$$\begin{cases} \mathsf{B}_{j,k}[\hat{d}_{j,k}] \triangleq \hat{d}_{j,k}\cdot(\hat{d}_{j,k}-1)\cdots(\hat{d}_{j,k}-B+1) \\ \mathsf{S}_k\Big[(\hat{d}_{j,k})_{j\in[\tilde{J}]}, (\hat{B}_{j,k})_{j\in[\tilde{J}]}\Big] \triangleq \sum_{j\in\tilde{J}} \hat{d}_{j,k}\cdot\hat{B}_{j,k} \end{cases} \tag{37}$$

One can check if $\omega \in [0, B^{\tilde{N}}-1]$ by checking if there exist vectors $\big((\hat{d}_{j,k})_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}, (\tilde{w}_k)_{k\in[\tilde{K}]}\big)$, such that

$$\begin{cases} \mathsf{B}_{j,k}[\hat{d}_{j,k}] \stackrel{?}{=} 0,\ \forall j\in[\tilde{J}], k\in[\tilde{K}] \\ \mathsf{S}_k\Big[(\hat{d}_{j,k})_{j\in[\tilde{J}]}, (\hat{B}_{j,k})_{j\in[\tilde{J}]}\Big] \stackrel{?}{=} \tilde{w}_k,\ \forall k\in[\tilde{K}] \\ \sum_{k\in[\tilde{K}]} \tilde{w}_k \stackrel{?}{=} \omega \end{cases} \tag{38}$$

To optimize verification, we utilize two levels of aggregation:

(1) We first aggregate the verification of $(\mathsf{B}_{j,k}[\cdot], \mathsf{S}_k[\cdot])_{k\in[\tilde{K}]}$ via batch verification. Let

$$\begin{aligned} \bar{d}_j[X] &\triangleq \mathsf{r}_j^{(\mathrm{d})}\cdot \mathsf{L}_0[X] + \textstyle\sum_{k\in[\tilde{K}]} \hat{d}_{j,k}\cdot \mathsf{L}_k[X] \\ \bar{B}_j[X] &\triangleq \textstyle\sum_{k\in[\tilde{K}]} \hat{B}_{j,k}\cdot \mathsf{L}_k[X] \\ \mathsf{B}_j[X] &\triangleq \frac{\bar{d}_j[X]\cdot(\bar{d}_j[X]-1)\cdots(\bar{d}_j[X]-B+1)}{\mathsf{L}_0[X]} \\ \mathsf{S}[X] &\triangleq \mathsf{s} + \frac{\sum_{k\in[\tilde{K}]} \tilde{w}_k\cdot \mathsf{L}_k[X] - \sum_{j\in[\tilde{J}]} \bar{d}_j[X]\cdot\bar{B}_j[X]}{\mathsf{L}_0[X]} \end{aligned}$$

where $\mathsf{B}_j[X]$ and $\mathsf{S}[X]$ are batched versions of $(\mathsf{B}_{j,k}[\cdot])_{k\in[\tilde{K}]}$ and $(\mathsf{S}_k[\cdot])_{k\in[\tilde{K}]}$, and $\mathsf{s}, \mathsf{r}_j^{(\mathrm{d})} \xleftarrow{\$} \mathbb{Z}_p$ are random masks. Suppose the prover commits $\big((\hat{d}_{j,k})_{j=1,k=1}^{\tilde{J}\ \ \tilde{K}}, (\tilde{w}_k)_{k\in[\tilde{K}]}\big)$ to $\big(D_k \triangleq \prod_{j\in[\tilde{J}]} G_j^{\hat{d}_{j,k}}\cdot Q^{\mathsf{r}_k^{(D)}}, \Omega_k \triangleq G^{\tilde{w}_k}\cdot Q^{\mathsf{r}_k^{(\Omega)}}\big)_{k\in[\tilde{K}]}$ in the beginning. Then the verifier can apply scenarios 2 and 3 to check $\mathsf{B}_j[X]$ and $\mathsf{S}[X]$, respectively.

(2) If we simply check $\mathsf{B}_j[X]$ for each $j \in [\tilde{J}]$, then it requires $\tilde{J}$ polynomial commitments. Thus, we aggregate the verification of $(\mathsf{B}_j[X])_{j\in[\tilde{J}]}$ via random linear combination that requires

only one polynomial commitment. Let

$$\mathtt{B}[X] \triangleq \sum_{j\in[\tilde{J}]} \beta^j \cdot \mathtt{B}_j[X]$$

where $\beta \xleftarrow{\$} \mathbb{Z}_p^*$ is a random challenge. Note that

$$\mathtt{B}[x] = \sum_{j\in[\tilde{J}]} \beta^j \cdot \mathtt{B}_j[x] = \frac{\sum_{j\in[\tilde{J}]} \beta^j \cdot \bar{d}_j[x]\cdot(\bar{d}_j[x]-1)\cdots(\bar{d}_j[x]-B+1)}{\mathsf{L}_0[x]}$$

where $x \xleftarrow{\$} \mathbb{Z}_p$ is a random challenge. Suppose the prover commits $\mathtt{B}[X]$ to $\mathtt{Cm}_\mathtt{B} \triangleq \mathtt{PolyCm}_{\mathtt{BCCGP}}[\mathtt{B}]$ and provides $\left(\pi_\mathtt{B} \triangleq \mathtt{PolyEv}_{\mathtt{BCCGP}}[\mathtt{B}, x], y_\mathtt{B} \triangleq \mathtt{B}[x], (\bar{d}_j \triangleq \bar{d}_j[x])_{j\in[\tilde{J}]}\right)$. Then the verifier can check $(\mathtt{B}_j[x])_{j\in[\tilde{J}]}$ by checking the following

$$\begin{cases} \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_\mathtt{B}, x, y_\mathtt{B}, \pi_\mathtt{B}] \stackrel{?}{=} 1 \\ \prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j} \cdot Q^{\tilde{\eta}_1} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (D_k)^{\mathsf{L}_k[x]} \cdot R_1^{\mathsf{L}_0[x]} \\ \sum_{j\in[\tilde{J}]} \beta^j \cdot \bar{d}_j \cdot (\bar{d}_j - 1)\cdots(\bar{d}_j - B + 1) \stackrel{?}{=} y_\mathtt{B} \cdot \mathsf{L}_0[x] \end{cases}$$

where $R_1 \triangleq \prod_{j\in[\tilde{J}]} G_j^{\mathsf{r}_j^{(\mathrm{d})}} \cdot Q^{\mathsf{r}_1}$ should be committed by the prover before knowing β and $\tilde{\eta}_1 \triangleq \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(D)} \cdot \mathsf{L}_k[x] + \mathsf{r}_1 \cdot \mathsf{L}_0[x]$ is provided by the prover.

5.2 Type-3 Range Argument Protocol

The full protocol of type-3 range argument is described in Fig. 5.

Theorem 5.1. VeRange *type-3 range argument protocol* Π_{ty3} *satisfies perfect completeness, SHVZK and CWE.*

The complete proof can be found in Appendix E.

Remarks: The degree of polynomial $\mathtt{B}_j[X]$ and $\mathtt{S}[X]$ are $(B-1)\tilde{K}$ and $2\tilde{K}$, respectively. Hence, BCCGP polynomial commitment verification takes around $2\sqrt{B\tilde{K}} + 2\sqrt{2\tilde{K}}$ group exponentiations. In addition, Step (44) takes $\tilde{J} + 2\tilde{K}$ group exponentiations. Since $\tilde{J}\tilde{K} \approx \tilde{N} = \frac{N}{\log B}$. The total number of group exponentiations is $2\sqrt{B\tilde{K}} + \tilde{J} + 2\tilde{K}$. To minimize group exponentiations, we set $\tilde{J} = \tilde{K} \approx (\frac{2N}{\log N})^{1/2}$ and $B \approx (\frac{N}{\log N})^{1/2}$. The resulting number of group exponentiations is $(3\sqrt{2}+2^{5/4})(\frac{N}{\log N})^{1/2}$. BCCGP polynomial commitment generation takes around $B\tilde{K} + \tilde{K}$ group exponentiations. It takes $\tilde{J}\tilde{K} + \tilde{J} + 3\tilde{K}$ group exponentiations for proving. The number of group exponentiations is $(2+\sqrt{2})\frac{N}{\log N} + 5\sqrt{2}(\frac{N}{\log N})^{1/2}$. The proof size includes $\sqrt{B\tilde{K}}$ group elements for BCCGP polynomial commitment and $2\tilde{K}$ group elements additionally, and $\sqrt{B\tilde{K}}$ field elements for BCCGP polynomial commitment and $\tilde{J}$ field elements additionally. In total, there are $3\sqrt{2}(\frac{N}{\log N})^{1/2}$ group elements and $2\sqrt{2}(\frac{N}{\log N})^{1/2}$ field elements.

5.3 Aggregating Type-3 Range Arguments

Multiple type-3 range arguments can be aggregated in a similar manner as type-2 in Sec. 4.3. Given $(\omega^{(t)})_{t\in[T]}$, the prover commits to $\mathtt{Cm}(\omega^{(t)}) \triangleq G^{\omega^{(t)}} \cdot Q^{\mathsf{r}_{\omega^{(t)}}}$ and aims to prove $\omega^{(t)} \in [0, B^{\tilde{N}} - 1]$ for all $t \in [T]$. Define $\left(\bar{\bar{d}}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$, $\left(\bar{\bar{B}}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$, $\left(\bar{\bar{w}}_{j,k} \triangleq \bar{\bar{d}}_{j,k} \cdot \bar{\bar{B}}_{j,k}\right)_{j\in[\tilde{J}],k\in[\tilde{K}]}$ the same manner as in Sec. 4.3.

Given a challenge $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, one can check if $\omega^{(t)} \in [0, B^{\tilde{N}} - 1]$ for all $t \in [T]$ via random linear combination by checking

$$\prod_{t\in[T]} \left(\mathtt{Cm}(\omega^{(t)})\right)^{\gamma^t} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \text{ and } \bar{\bar{w}}_{j,k} \stackrel{?}{\in} \{0, \bar{\bar{B}}_{j,k}, ..., (B\text{-}1)\cdot\bar{\bar{B}}_{j,k}\},$$

Figure 5: VeRange **type-3 range argument protocol**

$\Pi_{\mathrm{ty3}}\left[\mathtt{Cm}(\omega) \in \mathbb{G};\ \omega \in \mathbb{Z}_p, \mathsf{r}_\omega \in \mathbb{Z}_p^*\right]$

Setup : Distinct $z_0, z_1, ..., z_{\tilde{K}} \in \mathbb{Z}_p$

$$\mathsf{L}_k[X] \triangleq \prod_{k'\in\{0,...,\tilde{K}\}\setminus\{k\}} \frac{X - z_{k'}}{z_k - z_{k'}},\ \mathsf{L}_0[X] \triangleq \prod_{k\in[\tilde{K}]} (X - z_k),$$

$$\bar{B}_j[X] \triangleq \sum_{k\in[\tilde{K}]} \hat{B}_{j,k} \cdot \mathsf{L}_k[X]$$

$\mathcal{P}$: $\vec{\mathbf{d}} \in (\{0, ..., B-1\})^N$ is the B-ary digit decompo. of ω such that $\omega = \sum_{i\in[N]} d_i \cdot B^{i-1}$

$$\vec{\mathsf{r}}^{(\mathrm{d})} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{J}},\quad \vec{\mathsf{r}}^{(\Omega)}, \vec{\mathsf{r}}^{(D)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}},\quad \mathsf{s}, \mathsf{r}_1, \mathsf{r}_2 \xleftarrow{\$} \mathbb{Z}_p^*,\quad \mathsf{r}_{\tilde{K}}^{(\Omega)} \triangleq \mathsf{r}_\omega - \sum_{k\in[\tilde{K}-1]} \mathsf{r}_k^{(\Omega)}$$

$$\left(\hat{w}_{j,k} \triangleq \hat{d}_{j,k} \cdot \hat{B}_{j,k} \in \mathbb{Z}_p\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}},\quad \bar{d}_j[X] \triangleq \mathsf{r}_j^{(\mathrm{d})} \cdot \mathsf{L}_0[X] + \sum_{k\in[\tilde{K}]} \hat{d}_{j,k} \cdot \mathsf{L}_k[X]$$

$$\mathtt{B}_j[X] \triangleq \frac{\bar{d}_j[X] \cdot (\bar{d}_j[X] - 1)\cdots(\bar{d}_j[X] - B + 1)}{\mathsf{L}_0[X]}$$

$$\left(\tilde{w}_k \triangleq \sum_{j\in\tilde{J}} \hat{w}_{j,k}\right)_{k\in\tilde{K}},\quad \mathtt{S}[X] \triangleq \mathsf{s} + \frac{\sum_{k\in[\tilde{K}]} \tilde{w}_k \cdot \mathsf{L}_k[X] - \sum_{j\in[\tilde{J}]} \bar{d}_j[X] \cdot \bar{B}_j[X]}{\mathsf{L}_0[X]}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : \left(D_k \triangleq \prod_{j\in[\tilde{J}]} G_j^{\hat{d}_{j,k}} \cdot Q^{\mathsf{r}_k^{(D)}}\right)_{k\in[\tilde{K}]},\ R_1 \triangleq \prod_{j\in[\tilde{J}]} G_j^{\mathsf{r}_j^{(\mathrm{d})}} \cdot Q^{\mathsf{r}_1},\ R_2 \triangleq G^{\mathsf{s}} \cdot Q^{\mathsf{r}_2} \quad (39)$$

$$\left(\Omega_k \triangleq G^{\tilde{w}_k} \cdot Q^{\mathsf{r}_k^{(\Omega)}}\right)_{k\in[\tilde{K}]},\qquad \mathtt{Cm}_\mathtt{S} \triangleq \mathtt{PolyCm}_{\mathtt{BCCGP}}[\mathtt{S}] \in \mathbb{G}^{U+1} \quad (40)$$

$$\mathcal{P} \Leftarrow \mathcal{V} : \beta \xleftarrow{\$} \mathbb{Z}_p^*$$

$$\mathcal{P} : \mathtt{B}[X] \triangleq \sum_{j\in[\tilde{J}]} \beta^j \cdot \mathtt{B}_j[X]$$

$$\mathcal{P} \Rightarrow \mathcal{V} : \mathtt{Cm}_\mathtt{B} \triangleq \mathtt{PolyCm}_{\mathtt{BCCGP}}[\mathtt{B}] \in \mathbb{G}^{U+1}$$

$$\mathcal{P} \Leftarrow \mathcal{V} : x \xleftarrow{\$} \mathbb{Z}_p \setminus \{z_0, z_1, ..., z_{\tilde{K}}\}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : (\bar{d}_j \triangleq \bar{d}_j[x] \in \mathbb{Z}_p)_{j\in[\tilde{J}]},\quad y_\mathtt{B} \triangleq \mathtt{B}[x],\quad \pi_\mathtt{B} \triangleq \mathtt{PolyEv}_{\mathtt{BCCGP}}[\mathtt{B}, x] \in \mathbb{Z}_p^{V+1} \quad (41)$$

$$y_\mathtt{S} \triangleq \mathtt{S}[x] \in \mathbb{Z}_p,\quad \pi_\mathtt{S} \triangleq \mathtt{PolyEv}_{\mathtt{BCCGP}}[\mathtt{S}, x] \in \mathbb{Z}_p^{V+1} \quad (42)$$

$$\tilde{\eta}_1 \triangleq \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(D)} \cdot \mathsf{L}_k[x] + \mathsf{r}_1 \cdot \mathsf{L}_0[x],\ \tilde{\eta}_2 \triangleq \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(\Omega)} \cdot \mathsf{L}_k[x] + \mathsf{r}_2 \cdot \mathsf{L}_0[x] \quad (43)$$

$$\mathcal{V} : (\bar{B}_j \triangleq \bar{B}_j[x] \in \mathbb{Z}_p)_{j\in[\tilde{J}]}$$

Check

$$\begin{cases} \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_\mathtt{B}, x, y_\mathtt{B}, \pi_\mathtt{B}] \stackrel{?}{=} 1 \\ \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_\mathtt{S}, x, y_\mathtt{S}, \pi_\mathtt{S}] \stackrel{?}{=} 1 \\ \prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j} \cdot Q^{\tilde{\eta}_1} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (D_k)^{\mathsf{L}_k[x]} \cdot R_1^{\mathsf{L}_0[x]} \\ \sum_{j\in[\tilde{J}]} \beta^j \cdot \bar{d}_j \cdots (\bar{d}_j - B + 1) \stackrel{?}{=} y_\mathtt{B} \cdot \mathsf{L}_0[x] \\ G^{(y_\mathtt{S}\cdot\mathsf{L}_0[x] + \sum_{j\in[\tilde{J}]} \bar{d}_j \cdot \bar{B}_j)} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (\Omega_k)^{\mathsf{L}_k[x]} \cdot R_2^{\mathsf{L}_0[x]} \\ \mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \end{cases} \quad (44)$$

$\forall(j,k) \in [\tilde{J}] \times [\tilde{K}]$, where $(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \bar{w}_{j,k}} \cdot Q^{r_k^{(\Omega)}})_{k\in[\tilde{K}]}$ and the random masks $(r_k^{(\Omega)})_{k\in[\tilde{K}]}$ are set to satisfy $\sum_{t\in[T]} \gamma^t \cdot r_{\omega(t)} = \sum_{k\in[\tilde{K}]} r_k^{(\Omega)}$.

We can extend the type-3 range argument protocol to construct the full protocol of aggregated type-3 range argument, in a similar fashion as from the type-2 range argument protocol to the aggregated type-2 range argument protocol in Appendix G.2. The verification takes $O(\sqrt{\frac{TN}{\log(TN)}}) + T$ group exponentiations and proving takes $O(\frac{TN}{\log(TN)})$. The proof size includes $O(\sqrt{\frac{TN}{\log(TN)}})$ group elements and field elements.

The aggregated type-3 range argument protocol can be shown to satisfy perfect completeness, SHVZK and CWE, by extending Theorem 5.1 straightforwardly.

6 Empirical Evaluation

In this section, we provide an empirical evaluation of each type of VeRange and a comparison with the state-of-the-art range arguments (e.g., Bulletproofs, Bulletproof++, Flashproofs, SwiftRange, LLRing). Our experiments utilized the standard elliptic curve group *BN-128* on the Ethereum platform for both Pedersen commitment schemes and polynomial commitment schemes.

For the smart contract implementation on Ethereum, we rely on pre-compiled contract (EIP-196), which is limited to *BN-128* elliptic curve. To accommodate the commitment of a larger number ($\geq$ 128bit) on *BN-128* elliptic curve, we decompose the number into two smaller parts and prove the bit-decomposition of each part. For example, if $\omega \in [0, 2^{256}]$, then we let $\omega = \omega_1 \cdot 2^{128} + \omega_2$, where $\omega_1, \omega_2 \in [0, 2^{128}]$. A range argument of ω can be realized by proving the bit-decomposition of ω_1 and ω_2 separately[4].

6.1 Computational Overhead

We measured the runtime of proving and verification of Flashproofs, Bulletproofs, Bulletproofs++, SwiftRange, LLRing and VeRange, where the pre-computation optimization was applied to these arguments and the multi-exponentiation optimization was applied to the compression-friendly ones for fair efficiency comparisons. Note that we omit some measurement data for clarity. For a single argument, Table 2 shows all range arguments runtime of proving and verification in milliseconds. Fig. 6b and 6a graphically show the runtime of Flashproofs and VeRange. Fig. 7b illustrates the number of $\mathbb{G}$ exponentiations as this operation dominates the computational overhead. Our experimental results show that VeRange type-2 runs the fastest at 64bit and above, followed by VeRange type-3, type-2B, type-1 then Flashproofs, Bulletproofs++. Bulletproofs and SwiftRange compare unfavorably with all VeRange arguments in terms of verification computational efficiency. Specifically, type-2 achieves 1.56× and 1.94× proving efficiency for 64bit and 128bit ranges, respectively, as fast as Flashproofs. Type-1 runs 1.16× and 1.60× as fast as Flashproofs in verification for 64bit and 128bit ranges, respectively. All VeRange arguments are better than Flashproofs regarding the number of $\mathbb{G}$ exponentiations of proving, for bit length of range from 32bit. In addition, VeRange type-1 has the best performance in verification.

In Table 2, we compare the performance of aggregated Flashproofs and VeRange arguments for aggregating two and four 128bit range arguments (i.e., 2×128bit, 4×128bit). We observe that VeRange outperforms Flashproofs considerably in verification and proving time, when aggregating a large number of arguments

[4]Since all types of VeRange rely on bit or B-ary digit decomposition, they can also be used to prove a larger range ($\geq$ 128bit), despite the limitation of *BN-128*.

Table 2: Runtime (ms) comparison of VeRange and other range arguments

	N	32bit	64bit	128bit	256bit	512bit	2×128bit	4×128bit
Verification	Bulletproofs	187.2	355.9	673.3	1296.6	2533.2	-	-
	Bulletproof++	77.0	137.7	217.7	275.2	484.6	-	-
	SwiftRange	83.9	160.8	302.4	590.6	1151.0	-	-
	LLRing	958	1115	1337	-	-	-	-
	Flashproofs	27.1	35.5	65.0	88.6	150.9	93.0	166.0
	VeRange Type-1	24.9	30.7	40.7	56.8	78.8	58.9	85.1
	VeRange Type-2	33.6	43.9	54.0	68.4	87.3	61.9	84.0
	VeRange Type-2B	31.1	41.1	50.2	64.7	84.8	64.8	97.3
	VeRange Type-3	37.4	43.6	53.7	64.7	82.2	65.5	87.6
Proving	Bulletproofs	482.0	950.4	1885.8	3753.6	7487.3	-	-
	Bulletproof++	147.5	270.1	432.5	540.6	907.1	-	-
	SwiftRange	206.2	484.1	1020.2	2120.5	4286.1	-	-
	LLRing	19999	39586	83278	-	-	-	-
	Flashproofs	64.4	111.5	221.3	443.2	875.2	465.4	962.7
	VeRange Type-1	60	98.6	173.6	329.2	617.0	335.4	626.1
	VeRange Type-2	48.2	71.5	114.2	146.9	234.8	150.2	240.7
	VeRange Type-2B	55.4	98.3	143.0	275.9	436.4	254.2	487.0
	VeRange Type-3	52.3	71.6	116.0	179.6	380.8	175.1	412.2

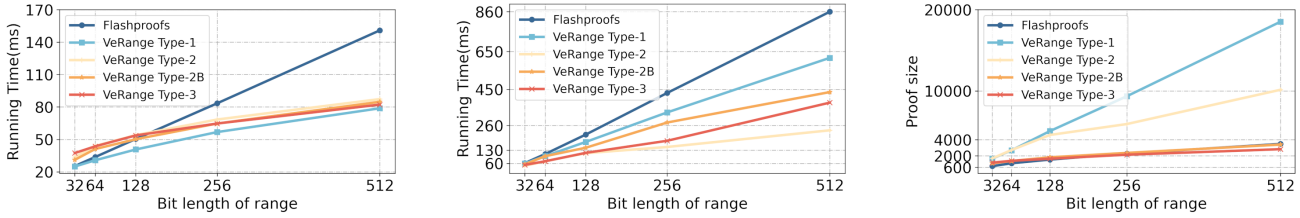
6.2 Communication Overhead

We assessed the proof sizes in bytes of a 256bit field. To optimize space utilization, we employed the compressed representation of elliptic curve points. This format consists of a 256bit value along with an additional bit indicating one of the two possible y coordinates. We provided line plots in Fig. 6c to demonstrate a more straightforward comparison of single arguments than Table 3. Bulletproof stands out as the most communication-efficient proof across various range sizes, including 64bit, and 128bit. The proof sizes grow logarithmically and square root-ly as N increases, whereas that of Flashproofs, VeRange type-1 and type-2 grows far quicker than SwiftRange and type-3. VeRange type-3 exhibits a slightly sharper growth in the proof size from the smallest 1168 bytes for 32bit as N grows. The proof size of type-3 increases as $O(\frac{N}{\log N})$.

Table 3: Proof size (byte), gas cost (Wei) and gas fee (USD$) comparison of VeRange and other range arguments

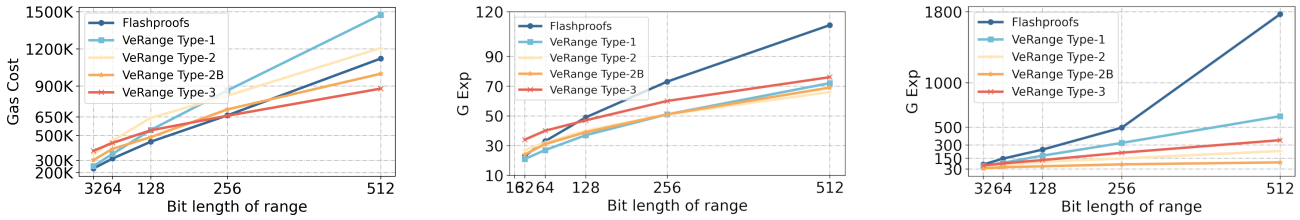
	N	32bit	64bit	128bit	256bit	512bit	2×128bit	4×128bit
Proof Size	Bulletproofs	610	674	739	804	868	-	-
	Bulletproof++	379	416	480	544	608	-	-
	SwiftRange	610	738	867	995	1123	-	-
	Flashproofs	738	1040	1544	2294	3472	2409	3819
	VeRange Type-1	1664	2688	5056	9344	18528	9811	20381
	VeRange Type-2	1696	2720	4576	5920	10144	6216	11158
	VeRange Type-2B	1088	1376	1824	2368	3392	2486	3731
	VeRange Type-3	1168	1395	1654	2144	2825	2251	3108
Gas Cost	Bulletproofs	2046K	3704K	5463K	7182K	9012K	-	-
	Bulletproof++	1364K	2170K	2952K	3903K	5006K	-	-
	SwiftRange	960K	2142K	3524K	4703K	5886K	-	-
	Flashproofs	233K	314K	450K	663K	1122K	696K	1234K
	VeRange Type-1	253K	351K	545K	864K	1475K	954K	1655K
	VeRange Type-2	347K	458K	643K	822K	1207K	873K	1313K
	VeRange Type-2B	301K	392K	487K	711K	999K	759K	1216K
	VeRange Type-3	376K	440K	542K	660K	879K	798K	1092K
Gas Fee	Bulletproofs	$45.0	$81.4	$120.1	$157.9	$198.1	-	-
	Bulletproof++	$30.0	$47.7	$64.9	$85.8	$110.0	-	-
	SwiftRange	$21.1	$47.1	$77.5	$103.4	$129.4	-	-
	Flashproofs	$5.1	$6.9	$9.9	$14.6	$24.7	$15.3	$27.2
	VeRange Type-1	$5.6	$7.7	$12.0	$19.0	$32.4	$21.0	$36.4
	VeRange Type-2	$7.6	$10.1	$14.1	$18.1	$26.5	$19.2	$28.9
	VeRange Type-2B	$6.6	$8.6	$10.7	$15.6	$22.0	$16.7	$26.7
	VeRange Type-3	$8.3	$9.7	$11.9	$14.5	$19.3	$17.5	$24.0

Note: Gas fees are calculated based on gas price and ETH/USD$ rate as 7 GWei and $3140 USD from [19] on 15 Apr 2024.



(a) Verification runtime (ms) (b) Proving runtime (ms) (c) Proof size (byte)

Figure 6: Runtime and proof size comparison



(a) Gas cost on smart contract (Wei) (b) Number of $\mathbb{G}$ exps in verification (c) Number of $\mathbb{G}$ exps in proving

Figure 7: Gas cost and number of $\mathbb{G}$ exps comparison

In Table 3, we compare the proof size and gas cost of aggregated Flashproofs and aggregated VeRange arguments for aggregating two and four 128bit range arguments. We observe that VeRange type-2B and type-3 attain lower proof sizes and gas costs, which confirms VeRange more suitable for aggregating multiple arguments.

6.3 Gas Costs

The estimation of gas costs is based on Solidity programming language and Truffle framework. For overall comparison of verification gas cost on smart contracts, refer to "Gas Cos" section in Table 3, which presents the gas costs of 32bit to 128bit ranges. Additionally, we present three pie charts in Fig. 8, providing a straightforward breakdown of gas costs for VeRange of different ranges.

In our evaluation, we computed gas costs for SwiftRange, Bulletproofs and VeRange by executing our implemented smart contracts. Notably, Bulletproofs incurred the highest gas consumption, significantly surpassing Flashproofs and VeRange, starting from 2046K for 32bit to 6144K for 128bit. Similarly, SwiftRange exhibited higher gas costs compared to VeRange, from 960K for 32bit to 4919K for 128bit. However, practical considerations lead us to conclude that Bulletproofs-like arguments and SwiftRange are less suitable for blockchain applications due to their gas inefficiency. Our VeRange, on the other hand, outperformed Flashproofs. Also, VeRange type-1 surpasses other three types of VeRange from 32bit to 64bit. Importantly, type-2B demonstrated increasing gas savings as bit size expanded. We also analyzed the breakdown of VeRange arguments' total gas cost, including initial cost, hashing, field operations, group exponentiation, and group multiplications. Fig. 8 and Table 4 illustrate this breakdown of 32bit to 128bit. They both clearly show that the group exponentiation dominates the gas cost. Conversely, field operations and group multiplications occupied relatively marginal proportions in both arguments. Type-1's efficiency shines through in group exponentiations, ultimately yielding comprehensive gas cost efficiency under 64bit range, then type-2B saves more gas above 128bit, and outperforms Flashproofs when the range size reaches to 512bit and above.

Table 4: Details of gas cost breakdown

	N	Deployment	Data storage	Hashing	Field ops	Group exps	Group muls	Total
32bit	Flashproofs	21K	27K	8K	21K	147K	8K	233K
	VeRange Type-1	21K	49K	10K	38K	126K	8K	253K
	VeRange Type-2	21K	56K	19K	82K	162K	7K	347K
	VeRange Type-2B	21K	37K	14k	63K	159K	8K	301K
	VeRange Type-3	21K	46K	11k	44K	245K	8K	376K
64bit	Flashproofs	21K	34K	10K	37K	201K	11K	314K
	VeRange Type-1	21K	73K	14K	67K	166K	10K	351K
	VeRange Type-2	21K	78K	22K	135K	192K	9K	458K
	VeRange Type-2B	21K	47K	21k	108K	186K	10K	392K
	VeRange Type-3	21K	53K	13K	56K	288K	10K	440K
128bit	Flashproofs	21K	44K	10K	90K	295K	14K	473K
	VeRange Type-1	21K	124K	17K	141K	224K	17K	545K
	VeRange Type-2	21K	122K	28K	219K	242K	11K	643K
	VeRange Type-2B	21K	56K	22k	139K	235K	13K	487K
	VeRange Type-3	21K	63K	16K	98K	332K	12K	542K

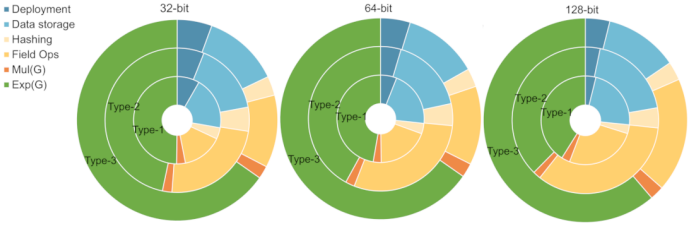


Figure 8: Breakdown of gas costs for VeRange.

References

[1] Sebastian Angel and Michael Walfish. 2013. Verifiable auctions for online ad exchanges. In *ACM SIGCOMM*. 195–206.

[2] James Bell, Adrià Gascón, Tancrède Lepoint, Baiyu Li, Sarah Meiklejohn, Mariana Raykova, and Cathie Yun. 2023. ACORN: Input validation for secure aggregation. In *USENIX Security)*. 4805–4822.

[3] Dan Boneh, Ben Fisch, Ariel Gabizon, and Zac Williamson. 2020. A Simple Range Proof From Polynomial Commitments. HackMD Markdown Knowledge Base. https://hackmd.io/@dabo/B1U4kx8XI.

[4] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. 2016. Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting. In *EUROCRYPT*.

[5] Jonathan Bootle and Jens Groth. 2018. Efficient Batch Zero-Knowledge Arguments for Low Degree Polynomials. In *PKC*. 561–588.

[6] Fabrice Boudot. 2000. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT*, Vol. 1807. 431–444.

[7] Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu. 2020. Zexe: Enabling decentralized private computation. In *IEEE SP*. 947–964.

[8] Benedikt Bunz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short Proofs for Confidential Transactions and More. In *IEEE SP*.

[9] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. 2008. Efficient protocols for set membership and range proofs. In *AsiaCrypt*.

[10] Konstantinos Chalkias, Shir Cohen, Kevin Lewi, Fredric Moezinia, and Yolan Romailler. 2021. HashWires: Hyperefficient Credential-Based Range Proofs. *roceedings on Privacy Enhancing Technologies* 4 (2021), 76–95.

[11] Sid Chi-Kin Chau and Yue Zhou. 2022. Blockchain-enabled decentralized privacy-preserving group purchasing for retail energy plans. In *Proc. ACM Intl. Conf. Future Energy Systems (e-Energy)*. 172–187.

[12] Miranda Christ, Foteini Baldimtsi, Konstantinos Kryptos Chalkias, Deepak Maram, Arnab Roy, and Joy Wang. 2024. SoK: Zero-Knowledge Range Proofs. Cryptology ePrint Archive, Paper 2024/430. https://eprint.iacr.org/2024/430

[13] Heewon Chung, Kyoohyung Han, Chanyang Ju, Myungsun Kim, and Jae Hong Seo. 2022. Bulletproofs+: Shorter Proofs for a Privacy-Enhanced Distributed Ledger. *IEEE Access* 10 (2022), 42081–42096.

[14] Geoffroy Couteau, Dahmun Goudarzi, Michael Klooß, and Michael Reichle. 2022. Sharp: Short Relaxed Range Proofs. In *ACM CCS*. 609–622.

[15] Geoffroy Couteau, Michael Klooß, Huang Lin, and Michael Reichle. 2021. Efficient range proofs with transparent setup from bounded integer commitments. In *Annual Intl. Conf. the Theory and Applications of Cryptographic Techniques*. Springer, 247–277.

[16] Gaby G Dagher, Benedikt Bünz, Joseph Bonneau, Jeremy Clark, and Dan Boneh. 2015. Provisions: Privacy-preserving proofs of solvency for bitcoin exchanges. In *ACM CCS*. 720–731.

[17] Samuel Dobson, Steven Galbraith, and Ben Smith. 2020. Trustless groups of unknown order with hyperelliptic curves. *Cryptology ePrint Archive - IACR* (2020).

[18] Liam Eagen, Sanket Kanjalkar, Tim Ruffing, and Jonas Nick. 2024. Bulletproofs++: Next Generation Confidential Transactions via Reciprocal Set Membership Arguments. In *EUROCRYPT*.

[19] Etherscan. 2024. https://etherscan.io/gasTracker.

[20] Amos Fiat and Adi Shamir. 1987. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*. 186–194.

[21] Ariel Gabizon, Zachary J Williamson, and Oana Ciobotaru. 2019. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *Cryptology ePrint Archive* (2019).

[22] Maxwell Gregory. 2016. Confidential Transactions. https://elementsproject.org/features/confidential-transactions/investigation.

[23] Jens Groth. 2005. Non-interactive Zero-Knowledge Arguments for Voting. In *Applied Cryptography and Network Security*. 467–482.

[24] Jens Groth. 2021. Non-interactive distributed key generation and key resharing. Cryptology ePrint Archive, Paper 2021/339. https://eprint.iacr.org/2021/339

[25] Xiangyu Hui and Sid Chi-Kin Chau. 2024. LLRing: Logarithmic Linkable Ring Signatures with Transparent Setup. In *ESORICS*.

[26] Jonathan Lee. 2021. Dory: Efficient, Transparent Arguments for Generalised Inner Products and Polynomial Commitments. In *TCC*. 1–34.

[27] Benoit Libert. 2023. Vector Commitments With Proofs of Smallness: Short Range Proofs and More. *Cryptology ePrint Archive* (2023).

[28] Lingjuan Lyu, Sid Chi-Kin Chau, Nan Wang, and Yifeng Zheng. 2022. Cloud-Based Privacy-Preserving Collaborative Consumption for Sharing Economy. *IEEE Trans. Cloud Computing* 10, 3 (2022), 1647–1660.

[29] Ronald L Rivest and Adi Shamir. 1996. PayWord and MicroMint: Two simple micropayment schemes. In *Intl. workshop on security protocols*. Springer, 69–87.

[30] Nan Wang and Sid Chi-Kin Chau. 2022. Flashproofs: Efficient Zero-Knowledge Arguments of Range and Polynomial Evaluation with Transparent Setup. In *AsiaCrypt*. 219–248.

[31] Nan Wang, Sid Chi-Kin Chau, and Dongxi Liu. 2024. SwiftRange: A Short and Efficient Zero-Knowledge Range Argument For Confidential Transactions and More. In *IEEE SP*.

[32] Nan Wang, Sid Chi-Kin Chau, and Yue Zhou. 2021. Privacy-Preserving Energy Storage Sharing with Blockchain. In *Proc. ACM Intl. Conf. Future Energy Systems (e-Energy)*. 185–198.

[33] Nan Wang, Sid Chi-Kin Chau, and Yue Zhou. 2021. Privacy-preserving energy storage sharing with blockchain and secure multi-party computation. *ACM SIGEnergy Energy Informatics Review* 1, 1 (2021), 32–50.

[34] Yue Zhou and Sid Chi-Kin Chau. 2021. Sharing Economy Meets Energy Markets: Group Purchasing of Energy Plans in Retail Energy Markets. In *ACM Intl. Conf. Systems for Energy-Efficient Built Environments (BuildSys)*.

[35] Yue Zhou and Sid Chi-Kin Chau. 2025. VeRange: Verification-efficient Zero-knowledge Range Arguments with Transparent Setup for Blockchain Applications and More. In *ACM AsiaCCS*.

[36] Hanwei Zhu, Sid Chi-Kin Chau, Gladhi Guarddin, and Weifa Liang. 2022. Integrating IoT-Sensing and Crowdsensing with Privacy: Privacy-Preserving Hybrid Sensing for Smart Cities. *ACM Trans. Internet-of-Things* 3, 4, Article 31 (Sep 2022), 30 pages.

Appendix

A Discussion and Conclusion

This paper presents VeRange, zero-knowledge range arguments in the discrete logarithm setting for blockchain deployment with a very low gas cost. In our evaluation, the majority of gas cost of the existing range arguments is attributed to the computational tasks, rather than the memory storage. On the other hand, all types of VeRange have a very low gas cost compared to Bulletproofs and SwiftRange. VeRange type-1 is well-suited for computation-critical applications, while Bulletproofs++, and VeRange type-3 are preferable for communication-critical scenarios. For 32-bit and smaller ranges, VeRange entirely outperforms Bulletproofs. For 64-bit ranges, VeRange type-1, type-2, and type-2B have a significant advantage over Bulletproofs and Flashproofs for verification efficiency. For ranges larger than 64-bit, VeRange type-1 is the optimal choice, for even larger than 256-bit, type-2 becomes the best when computational efficiency takes precedence over communication efficiency. VeRange type-3 outperforms Flashproofs in gas cost and proof size for above 256-bit ranges. Furthermore, if we consider aggregating multiple range arguments, VeRange type-2B and type-3 provide better aggregation gains, because of their lower asymptotic order of magnitude. Hence, VeRange type-2B and type-3 are the most cost-effective solution for blockchain applications among the recent discrete logarithmic range arguments for a sufficiently large N. Overall, VeRange incurs merely $\sim 10\%$ of the gas cost of Bulletproofs.

In future work, we will apply VeRange to a wide range of privacy-preserving blockchain-enabled applications [11, 28, 32–34, 36].

B Additional Definitions

Denote a polynomial-time decidable tertiary relation by $\mathscr{R} \subset \{0,1\}^{*3}$. A language dependent on pp is defined as $\mathscr{L}_{\mathscr{R}}^{\mathsf{pp}} \triangleq \{x \mid \exists \omega : (\mathsf{pp}, x, \omega) \in \mathscr{R}\}$, where ω is a witness for a statement x in the relation $(\mathsf{pp}, x, \omega) \in \mathscr{R}$.

Definition B.1 (Argument of Knowledge). Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is called an *argument of knowledge* for relation $\mathscr{R}$, if it satisfies the perfect completeness (Definition (B.2)) and Computational Witness-Extended Emulation (Definition (B.3)).

Definition B.2 (Completeness). Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies *completeness*, if for any PPT adversary $\mathcal{A}$:

$$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Accept}[\mathsf{tr}]\\ =1\end{array} \;\middle|\; \begin{array}{c}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda),\\ (\mathsf{pp}, x, \omega) \in \mathscr{R},\\ \mathsf{tr} \leftarrow \langle \mathcal{P}(\mathsf{pp}, x, \omega), \mathcal{V}(\mathsf{pp}, x)\rangle\end{array}\right] \geq 1 - \mathrm{negl}(\lambda)$$

We call it *perfect completeness*, if $\mathrm{negl}(\lambda) = 0$.

Definition B.3 (Computational Witness-Extended Emulation (CWE) [8]). Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies CWE, if there exists an expected polynomial-time emulator $\mathcal{E}$, such that for any interactive adversaries $\mathcal{A}_1, \mathcal{A}_2$:

$$\left|\mathsf{Pr}\left[\mathcal{A}_1[\mathsf{tr}] = 1 \;\middle|\; \begin{array}{c}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda),\\ (x, \tilde{w}, \tilde{\mathcal{P}}) \leftarrow \mathcal{A}_2[\mathsf{pp}],\\ \mathsf{tr} \leftarrow \langle \tilde{\mathcal{P}}(\mathsf{pp}, x, \tilde{w}), \mathcal{V}(\mathsf{pp}, x)\rangle\end{array}\right]\right.$$
$$\left. -\mathsf{Pr}\left[\begin{array}{c}\mathcal{A}_1[\mathsf{tr}'] = 1\\ \wedge(\mathsf{Accept}[\mathsf{tr}'] = 1 \Rightarrow\\ (\mathsf{pp}, x, w') \in \mathscr{R})\end{array} \;\middle|\; \begin{array}{c}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda),\\ (x, \tilde{w}, \tilde{\mathcal{P}}) \leftarrow \mathcal{A}_2[\mathsf{pp}],\\ (\mathsf{tr}', w') \leftarrow \mathcal{E}^{O}[\mathsf{pp}, x]\end{array}\right]\right| \leq \mathrm{negl}(\lambda)$$

where $\tilde{\mathcal{P}}$ is a deterministic polynomial-time algorithm, $\mathcal{A}_1[\mathsf{tr}]$ recognizes the transcripts that are produced by $\tilde{\mathcal{P}}$, and $\mathcal{O}$ is a rewindable oracle that can rewind the transcript $\langle\tilde{\mathcal{P}}(\mathsf{pp}, x, \tilde{w}), \mathcal{V}(\mathsf{pp}, x)\rangle$ and control the randomness in $\mathcal{V}$.

Definition B.4 (Public Coin). Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is called *public-coin*, if the verifier chooses her messages uniformly at random, independent from the messages sent by the prover. Let e be the public-coin challenge. The transcript of a public-coin argument system is defined as $\mathsf{tr} = \langle\mathcal{P}(\mathsf{pp}, x, \omega), \mathcal{V}(\mathsf{pp}, x; e)\rangle$.

Definition B.5 (Special Honest-Verifier Zero-Knowledge (SHVZK)). A public-coin argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies SHVZK, if there exists an efficient simulator $\mathcal{S}$, such that for any PPT adversary $\mathcal{A}$:

$$\left|\mathsf{Pr}\left[\begin{array}{c}\mathtt{Accept}[\mathsf{tr}]\\=1\\\wedge\ (\mathsf{pp},x,\omega)\in\mathscr{R}\end{array}\middle|\begin{array}{c}\mathsf{pp}\leftarrow\mathcal{G}(1^\lambda),\\(x,\omega,e)\leftarrow\mathcal{A}[\mathsf{pp}],\\\mathsf{tr}\leftarrow\langle\mathcal{P}(\mathsf{pp},x,\omega),\mathcal{V}(\mathsf{pp},x;e)\rangle\end{array}\right]\right.$$
$$\left.-\mathsf{Pr}\left[\begin{array}{c}\mathtt{Accept}[\mathsf{tr}]\\=1\\\wedge\ (\mathsf{pp},x,\omega)\in\mathscr{R}\end{array}\middle|\begin{array}{c}\mathsf{pp}\leftarrow\mathcal{G}(1^\lambda),\\(x,\omega,e)\leftarrow\mathcal{A}[\mathsf{pp}],\\\mathsf{tr}\leftarrow\mathcal{S}[\mathsf{pp},x;e]\end{array}\right]\right|\leq\mathrm{negl}(\lambda)$$

Definition B.6 (Fiat-Shamir Transformation). A multi-move interactive public-coin argument of knowledge can be converted to a non-interactive argument of knowledge by replacing the public-coin challenges by the output of a cryptographic hash function, which produces seemingly random output and is regarded as a replacement for a verifier.

C Proofs for VeRange Type-1 Range Argument

Theorem C.1. VeRange *type-1 range argument protocol* Π_{ty1} *satisfies perfect completeness, SHVZK and CWE.*

Proof. **Perfect Completeness**: Π_{ty1} satisfies perfect completeness by following Eqns. (3)-(6).

SHVZK: We define a simulator as follows. First, the simulator generates group elements $\big((W_k)_{k\in[K-1]}, (T_k)_{k\in[K]}\big)$ at Step (7) and field elements $\big((v_{j,k})_{j\in[J],k\in[K]}, \eta_1, \eta_2\big)$ at Step (10) at uniformly random. Then the simulator sets

$$W_K \triangleq \mathtt{Cm}(\omega)\cdot\Big(\prod_{k\in[K-1]} W_k\Big)^{-1}$$

Next, after learning the challenge $\vec{\epsilon}$ from the verifier, the simulator rewinds to Step (8) to set

$$\begin{cases} S \triangleq \prod_{j\in[J]} H_j^{\sum_{k\in[K]} v_{j,k}\cdot u_{j,k}}\cdot Q^{\eta_1}\cdot\Big(\prod_{k\in[K]} T_k^{\epsilon_k}\Big)^{-1}\\ R \triangleq G^{\sum_{j\in[J],k\in[K]} v_{j,k}}\cdot Q^{\eta_2}\cdot\Big(\prod_{k\in[K]} W_k^{\epsilon_k}\Big)^{-1}\end{cases}$$

One can check that the above settings of $\big((W_k, T_k)_{k\in[K]}, R, S\big)$ and $\big((v_{j,k})_{j\in[J],k\in[K]}, \eta_1, \eta_2\big)$ can successfully pass the verification at the verifier without the witness ω. Moreover, the transcripts appear to be uniformly random.

CWE: We define an emulator as follows. The emulator emulates the prover with random challenge $\vec{\epsilon} \xleftarrow{\$} \mathbb{Z}_p^{*K}$ to generate a transcript and if the transcript is accepting it rewinds in the protocol with new different challenges until it has generated L different accepting arguments. If the prover probabilistically produces an accepting argument with probability δ, then we expect the emulator to rewind $\mathsf{L}\cdot\frac{1}{\delta}$ times to obtain L accepting arguments. The emulator also emulates the prover's probability δ for producing an accepting argument. Hence, the emulator is expected to rewind $\delta\cdot\frac{\mathsf{L}}{\delta} = \mathsf{L}$ times, which runs in expected polynomial time. Next, we describe how to extract a valid witness to statement $\omega \in [0, 2^N]$ from L accepting arguments.

Given the initial message $\big((W_k, T_k)_{k\in[K]}, R, S\big)$ from the prover in an honest execution of Π_{ty1}, we rewind L times to Step (9) with L different random challenges $\big(\vec{\epsilon}^{\,(\ell)} \xleftarrow{\$} \mathbb{Z}_p^{*K}\big)_{\ell\in[\mathsf{L}]}$ to obtain transcripts $\big((v_{j,k}^{(\ell)}, u_{j,k}^{(\ell)})_{j\in[J],k\in[K]}, \eta_1^{(\ell)}, \eta_2^{(\ell)}\big)_{\ell\in[\mathsf{L}]}$ which the verifier checks at Step (11) to satisfy the following for each $\ell\in[\mathsf{L}]$:

$$\prod_{j\in[J]} H_j^{\sum_{k\in[K]} v_{j,k}^{(\ell)}\cdot u_{j,k}^{(\ell)}}\cdot Q^{\eta_1^{(\ell)}} = \prod_{k\in[K]} T_k^{\epsilon_k^{(\ell)}}\cdot S \text{ and}$$
$$G^{\sum_{j\in[J],k\in[K]} v_{j,k}^{(\ell)}}\cdot Q^{\eta_2^{(\ell)}} = \prod_{k\in[K]} W_k^{\epsilon_k^{(\ell)}}\cdot R \tag{45}$$

Suppose Eqns. (45) are satisfied for all $\ell\in[\mathsf{L}]$. We extract $\big(\mathsf{m}_R, \mathsf{r}'_R, \mathsf{r}'_S, (w'_k, \mathsf{r}'^{(W)}_k, \mathsf{r}'^{(T)}_k)_{k\in[K]}, (\mathsf{m}_j^{(S)})_{j\in[J]}, (t'_{j,k})_{j\in[J],k\in[K]}\big)$ from Eqns.(45) by

$$\begin{pmatrix}\sum_{k\in[K]} v_{1,k}^{(1)}\cdot u_{1,k}^{(1)} & \cdots & \sum_{k\in[K]} v_{J,k}^{(1)}\cdot u_{J,k}^{(1)} & \eta_1^{(1)}\\ \vdots & \ddots & \vdots & \vdots\\ \sum_{k\in[K]} u_{1,k}^{(\mathsf{L})}\cdot v_{1,k}^{(\mathsf{L})} & \cdots & \sum_{k\in[K]} v_{J,k}^{(\mathsf{L})}\cdot u_{J,k}^{(\mathsf{L})} & \eta_1^{(\mathsf{L})}\end{pmatrix}$$
$$=\begin{pmatrix}\epsilon_1^{(1)} & \cdots & \epsilon_K^{(1)} & 1\\ \vdots & \ddots & \vdots & \vdots\\ \epsilon_1^{(\mathsf{L})} & \cdots & \epsilon_K^{(\mathsf{L})} & 1\end{pmatrix}\cdot\begin{pmatrix}t'_{1,1} & \cdots & t'_{J,1} & \mathsf{r}'^{(T)}_1\\ \vdots & \ddots & \vdots & \vdots\\ t'_{1,K} & \cdots & t'_{J,K} & \mathsf{r}'^{(T)}_K\\ \mathsf{m}_1^{(S)} & \cdots & \mathsf{m}_J^{(S)} & \mathsf{r}'_S\end{pmatrix} \tag{46}$$

$$\begin{pmatrix}\sum_{j\in[J],k\in[K]} v_{j,k}^{(1)} & \eta_2^{(1)}\\ \vdots & \vdots\\ \sum_{j\in[J],k\in[K]} v_{j,k}^{(\mathsf{L})} & \eta_2^{(\mathsf{L})}\end{pmatrix}$$
$$=\begin{pmatrix}\epsilon_1^{(1)} & \cdots & \epsilon_K^{(1)} & 1\\ \vdots & \ddots & \vdots & \vdots\\ \epsilon_1^{(\mathsf{L})} & \cdots & \epsilon_K^{(\mathsf{L})} & 1\end{pmatrix}\cdot\begin{pmatrix}w'_1 & \mathsf{r}'^{(W)}_1\\ \vdots & \vdots\\ w'_K & \mathsf{r}'^{(W)}_K\\ \mathsf{m}_R & \mathsf{r}'_R\end{pmatrix} \tag{47}$$

If $\mathsf{L} = K+1$ and $\big(\vec{\epsilon}^{\,(\ell)} \xleftarrow{\$} \mathbb{Z}_p^{*K}\big)_{\ell\in[\mathsf{L}]}$ are chosen at uniformaly random, then the probability that the following matrix E:

$$\mathsf{E} \triangleq \begin{pmatrix}\epsilon_1^{(1)} & \cdots & \epsilon_K^{(1)} & 1\\ \vdots & \ddots & \vdots & \vdots\\ \epsilon_1^{(\mathsf{L})} & \cdots & \epsilon_K^{(\mathsf{L})} & 1\end{pmatrix}$$

has zero determinant is negligible in the size of $\mathbb{Z}_q$ by an argument based on Schwartz–Zippel Lemma.

Hence, E is invertible with high probability and $\big(\mathsf{m}_R, \mathsf{r}'_R, \mathsf{r}'_S, (w'_k, \mathsf{r}'^{(W)}_k, \mathsf{r}'^{(T)}_k)_{k\in[K]}, (\mathsf{m}_j^{(S)})_{j\in[J]}, (t'_{j,k})_{j\in[J],k\in[K]}\big)$

can be uniquely determined by multiplying E^{-1} to both sides of Eqns. (46)-(47).

Note that if we let $\big((W_k, T_k)_{k\in[K]}, R, S\big)$ by the following

$$W_k = G^{w'_k} \cdot Q^{r'^{(W)}_k}, \quad T_k = \prod_{j\in[J]} H_j^{t'_{j,k}} \cdot Q^{r'^{(T)}_k}$$

$$R = G^{m_R} \cdot Q^{r'_R}, \quad S = \prod_{j\in[J]} H_j^{m^{(S)}_j} \cdot Q^{r'_S}$$

then by Eqns. (46)-(47) $\big((W_k, T_k)_{k\in[K]}, R, S\big)$ can satisfy Eqn. (45) as follows:

$$\prod_{j\in[J]} H_j^{\sum_{k\in[K]} v^{(\ell)}_{j,k}\cdot u^{(\ell)}_{j,k}} \cdot Q^{\eta^{(\ell)}_1} = \prod_{k\in[K]} \Big(\prod_{j\in[J]} H_j^{t'_{j,k}} \cdot Q^{r'^{(T)}_k}\Big)^{\epsilon^{(\ell)}_k} \cdot \prod_{j\in[J]} H_j^{m^{(S)}_j} \cdot Q^{r'_S}$$
$$= \prod_{k\in[K]} T_k^{\epsilon^{(\ell)}_k} \cdot S \tag{48}$$

$$G^{\sum_{j\in[J],k\in[K]} v^{(\ell)}_{j,k}} \cdot Q^{\eta^{(\ell)}_2} = \prod_{k\in[K]} \big(G^{w'_k} \cdot Q^{r'^{(W)}_k}\big)^{\epsilon^{(\ell)}_k} \cdot G^{m_R} \cdot Q^{r'_R}$$
$$= \prod_{k\in[K]} W_k^{\epsilon^{(\ell)}_k} \cdot R \tag{49}$$

Namely, $\big(m_R, r'_R, r'_S, (w'_k, r'^{(W)}_k, r'^{(T)}_k)_{k\in[K]}, (m^{(S)}_j)_{j\in[J]}, (t'_{j,k})_{j\in[J],k\in[K]}\big)$ are satisfying witness.

Next, pick $\vec{\epsilon}^{\,(1)}$ and $\vec{\epsilon}^{\,(2)}$ from $\{\vec{\epsilon}^{\,(\ell)}\}_{\ell\in[\mathsf{L}]}$. Note that there is negligible probability in the size of $\mathbb{Z}_q$ that $\epsilon^{(1)}_k = \epsilon^{(2)}_k$ for any k. One can extract witness $(w'_{j,k}, r'_{j,k})$ by solving the following

$$v^{(1)}_{j,k} = w'_{j,k}\cdot\epsilon^{(1)}_k + r'_{j,k}, \qquad v^{(2)}_{j,k} = w'_{j,k}\cdot\epsilon^{(2)}_k + r'_{j,k}$$

We substitute $(w'_{j,k}, r'_{j,k})_{j\in[J],k\in[K]}$ into $(v^{(1)}_{j,k}, u^{(1)}_{j,k})_{j\in[J],k\in[K]}$ by

$$\big(v^{(1)}_{j,k} = w'_{j,k}\cdot\epsilon^{(1)}_k + r'_{j,k}\big)_{j\in[J],k\in[K]}, \big(u^{(1)}_{j,k} = \hat{2}_{j,k}\cdot\epsilon^{(1)}_k - v^{(j,1)}_k\big)_{j\in[J],k\in[K]}$$

We then obtain

$$\prod_{j\in[J]} H_j^{\sum_{k\in[K]} v^{(1)}_{j,k}\cdot u^{(1)}_{j,k}} \cdot Q^{\eta^{(1)}_1}$$
$$= \prod_{j\in[J]} H_j^{\sum_{k\in[K]} (\hat{2}_{j,k}-w'_{j,k})w'_{j,k}\cdot(\epsilon^{(1)}_k)^2 + r'_{j,k}(\hat{2}_{j,k}-2w'_{j,k})\cdot\epsilon^{(1)}_k - (r'_{j,k})^2} \cdot Q^{\eta^{(1)}_1} \tag{50}$$

We compare Eqn. (50) with Eqn. (48) (setting $\ell = 1$). Based on the DLR assumption, we obtain the exponent of each H_j in Eqn. (50) and Eqn. (48) as

$$\sum_{k\in[K]} (\hat{2}_{j,k} - w'_{j,k})w'_{j,k}\cdot(\epsilon^{(1)}_k)^2 + r'_{j,k}(\hat{2}_{j,k} - 2w'_{j,k})\cdot\epsilon^{(1)}_k - (r'_{j,k})^2$$
$$= \sum_{k\in[K]} t'_{j,k}\epsilon^{(1)}_k + m^{(S)}_j \tag{51}$$

Note that $\vec{\epsilon}^{\,(1)}$ is selected at uniformly random. Hence, Eqn. (51) is always true with high probability, only if the coefficients of $(\epsilon^{(1)}_k)^2$ and $\epsilon^{(1)}_k$ for all $k \in [K]$ and constant terms of LHS and RHS are equivalent, namely,

$$(\hat{2}_{j,k} - w'_{j,k})w'_{j,k} = 0 \quad \text{and} \quad r'_{j,k}(\hat{2}_{j,k} - 2w'_{j,k}) = (t'_{j,k})$$

Hence, this implies $w'_{j,k} \in \{0, \hat{2}_{j,k}\}$.

Also, we substitute $(w'_{j,k}, r'_{j,k})_{j\in[J],k\in[K]}$ into $(v^{(1)}_{j,k})_{j\in[J],k\in[K]}$ to obtain

$$G^{\sum_{j\in[J],k\in[K]} v^{(\ell)}_{j,k}} \cdot Q^{\eta^{(\ell)}_2} = G^{\sum_{j\in[J],k\in[K]} w'_{j,k}\cdot\epsilon^{(1)}_k + r'_{j,k}} \cdot Q^{\eta^{(\ell)}_2} \tag{52}$$

We compare Eqn. (52) with Eqn. (49) (setting $\ell = 1$). By equating the coefficient of $\epsilon^{(1)}_k$ for all $k \in [K]$, we obtain

$$\sum_{j\in[J]} w'_{j,k} = w'_k$$

Finally, we obtain

$$G^{\omega}\cdot Q^{r_\omega} = \mathtt{Cm}(\omega) = \prod_{k\in[K]} W_k = \prod_{k\in[K]} G^{w'_k}\cdot Q^{r'^{(W)}_k}$$
$$= G^{\sum_{k\in[K]} w'_k} \cdot Q^{\sum_{k\in[K]} r'^{(W)}_k} = G^{\sum_{k\in[K]}\sum_{j\in[J]} w'_{j,k}} \cdot Q^{\sum_{k\in[K]} r'^{(W)}_k} \tag{53}$$

By the DLR assumption, we obtain $\omega = \sum_{k\in[K]}\sum_{j\in[J]} w'_{j,k}$. Therefore, this proves that the extracted $(w'_{j,k} \in \{0, \hat{2}_{j,k}\})_{j\in[J],k\in[K]}$ is a valid witness to $\omega \in [0, 2^N]$. □

D Proofs for VeRange Type-2 Range Argument

Theorem D.1. VeRange *type-2 range argument protocol* Π_{ty2} *satisfies perfect completeness, SHVZK and CWE.*

Proof. **Perfect Completeness**: Π_{ty2} satisfies perfect completeness by following Eqns. (21)-(24).

SHVZK: We define a simulator as follows. First, the simulator generates group elements $\big((\Omega_k)_{k\in[\tilde{K}-1]}, (V_k, \tilde{T}_k)_{k\in[\tilde{K}]}, \{M_c\}_{c=1}^{B-1}\big)$ at Steps (25)-(27) and field elements $\big((\nu_{j,k}, \mu_{j,k})_{(j,k)\in\mathcal{B}}, \tilde{\eta}_1, \tilde{\eta}_2\big)$ at Steps (30)-(31) at uniformly random. Then the simulator sets

$$\Omega_{\tilde{K}} \triangleq \mathtt{Cm}(\omega)\cdot\Big(\prod_{k\in[\tilde{K}-1]}\Omega_k\Big)^{-1}$$

Next, after learning the challenge $\vec{\epsilon}$ from the verifier, the simulator rewinds to Step (27) to set

$$\begin{cases} \tilde{S} \triangleq \sum_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]}\tilde{\nu}_{j,k}\cdot\tilde{\mu}_{j,k}}\cdot Q^{\tilde{\eta}_1}\cdot\Big(\big(\prod_{j\in[\tilde{J}]} H_j\big)^{\sum_{k\in[\tilde{K}]}\epsilon^2_k}\cdot\prod_{k\in[\tilde{K}]}\tilde{T}_k^{\epsilon_k}\Big)^{-1} \\ M_0 \triangleq G^{\sum_{(j,k)\in\mathcal{B}}\mu'_{j,k}}\cdot Q^{\tilde{\eta}_2}\cdot\Big(\prod_{c=1}^{B-1} M_c^{\frac{1}{\alpha+c}}\cdot\prod_{k\in[\tilde{K}]} V_k^{\epsilon_k^{-1}}\Big)^{-1} \\ \tilde{R} \triangleq G^{\sum_{(j,k)\in\mathcal{B}}\nu_{j,k}}\cdot Q^{\tilde{\eta}_3}\cdot\Big(\prod_{k\in[\tilde{K}]}\Omega_k^{\epsilon_k}\Big)^{-1} \end{cases}$$

One can check that the above settings of $\big((\Omega_k, V_k, \tilde{T}_k)_{k\in[\tilde{K}]}, \{M_c\}_{c=0}^{B-1}, \tilde{R}, \tilde{S}\big)$ and $\big((\nu_{j,k}, \mu_{j,k})_{(j,k)\in\mathcal{B}}, \tilde{\eta}_1, \tilde{\eta}_2, \tilde{\eta}_3\big)$ can successfully pass the verification at the verifier without the witness ω. Moreover, the transcripts appear to be uniformly random.

CWE: We define an emulator as follows in a similar manner as in Theorem C.1. The emulator emulates a probabilistic prover with a number of random challenges and rewinding to generate multiple accepting arguments. The emulator is expected to run in expected polynomial time. Next we describe how to extract a valid witness to statement $\omega \in [0, 2^{\tilde{N}}]$.

Given the initial message $\big((\Omega_k, V_k, \tilde{T}_k)_{k\in[\tilde{K}]}, \{M_c\}_{c=0}^{B-1}, \tilde{R}, \tilde{S}\big)$ from the prover in an honest execution of Π_{ty2}, we proceed to a 2-stage rewinding:

(1) *Stage 1*: We first obtain a random challenge $\alpha^{(1)} \xleftarrow{\$} \mathbb{Z}_p^*$. Then, we rewind L times to Step (29) to obtain different random challenges $\big(\vec{\epsilon}^{\,(1,\ell)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}\big)_{\ell\in[\mathsf{L}]}$. For each challenge tuple $(\alpha^{(1)}, \vec{\epsilon}^{\,(1,\ell)})$, we obtain accepting transcript $\big((v_{j,k}^{(1,\ell)}, \mu_{j,k}^{(1,\ell)})_{(j,k)\in\mathcal{B}}, \tilde{\eta}_1^{(1,\ell)}, \tilde{\eta}_2^{(1,\ell)}, \tilde{\eta}_3^{(1,\ell)}\big)$ which the verifier checks at Step (32) to satisfy the following:

$$\begin{cases} \prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]} \tilde{v}_{j,k}^{(1,\ell)}\cdot\tilde{\mu}_{j,k}^{(1,\ell)}} \cdot Q^{\tilde{\eta}_1^{(1,\ell)}} \stackrel{?}{=} \big(\prod_{j\in[\tilde{J}]} H_j\big)^{\sum_{k\in[\tilde{K}]}(\epsilon_k^{(1,\ell)})^2} \cdot \prod_{k\in[\tilde{K}]} \tilde{T}_k^{\epsilon_k^{(1,\ell)}} \cdot \tilde{S} \\ G^{\sum_{(j,k)\in\mathcal{B}} v_{j,k}^{(1,\ell)}} \cdot Q^{\tilde{\eta}_3^{(1,\ell)}} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k^{\epsilon_k^{(1,\ell)}} \cdot \tilde{R} \end{cases} \tag{54}$$

Next, we extract $\big(\mathsf{m}_R, \mathsf{r}'_R, \mathsf{r}'_S, (\tilde{w}'_k, \mathsf{r}'^{(\Omega)}_k, \mathsf{r}'^{(T)}_k)_{k\in[\tilde{K}]}, (\mathsf{m}_j^{(S)})_{j\in[\tilde{J}]}, (\tau'_{j,k})_{j=1,k=1}^{\tilde{J}\ \tilde{K}}\big)$ by

$$\begin{pmatrix} \sum_{k\in[\tilde{K}]} \tilde{v}_{1,k}^{(1,1)}\cdot\tilde{\mu}_{1,k}^{(1,1)} - \sum_{k\in[\tilde{K}]}(\epsilon_k^{(1,1)})^2 & \cdots & \sum_{k\in[\tilde{K}]} \tilde{v}_{\tilde{J},k}^{(1,1)}\cdot\tilde{\mu}_{\tilde{J},k}^{(1,1)} - \sum_{k\in[\tilde{K}]}(\epsilon_k^{(1,1)})^2 & \tilde{\eta}_1^{(1,1)} \\ \vdots & \ddots & \vdots & \vdots \\ \sum_{k\in[\tilde{K}]} \tilde{v}_{1,k}^{(1,\mathsf{L})}\cdot\tilde{\mu}_{1,k}^{(1,\mathsf{L})} - \sum_{k\in[\tilde{K}]}(\epsilon_k^{(1,\mathsf{L})})^2 & \cdots & \sum_{k\in[\tilde{K}]} \tilde{v}_{\tilde{J},k}^{(1,\mathsf{L})}\cdot\tilde{\mu}_{\tilde{J},k}^{(1,\mathsf{L})} - \sum_{k\in[\tilde{K}]}(\epsilon_k^{(1,\mathsf{L})})^2 & \tilde{\eta}_1^{(1,\mathsf{L})} \end{pmatrix}$$

$$= \begin{pmatrix} \epsilon_1^{(1,1)} & \cdots & \epsilon_{\tilde{K}}^{(1,1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \epsilon_1^{(1,\mathsf{L})} & \cdots & \epsilon_{\tilde{K}}^{(1,\mathsf{L})} & 1 \end{pmatrix} \cdot \begin{pmatrix} \tau'_{1,1} & \cdots & \tau'_{\tilde{J},1} & \mathsf{r}'^{(T)}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \tau'_{1,\tilde{K}} & \cdots & \tau'_{\tilde{J},\tilde{K}} & \mathsf{r}'^{(T)}_{\tilde{K}} \\ \mathsf{m}_1^{(S)} & \cdots & \mathsf{m}_{\tilde{J}}^{(S)} & \mathsf{r}'_S \end{pmatrix} \tag{55}$$

$$\begin{pmatrix} \sum_{(j,k)\in\mathcal{B}} v_{j,k}^{(1,\ell)} & \tilde{\eta}_3^{(1,1)} \\ \vdots & \vdots \\ \sum_{(j,k)\in\mathcal{B}} v_{j,k}^{(1,\ell)} & \tilde{\eta}_3^{(1,\mathsf{L})} \end{pmatrix} = \begin{pmatrix} \epsilon_1^{(1,1)} & \cdots & \epsilon_{\tilde{K}}^{(1,1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \epsilon_1^{(1,\mathsf{L})} & \cdots & \epsilon_{\tilde{K}}^{(1,\mathsf{L})} & 1 \end{pmatrix} \cdot \begin{pmatrix} \tilde{w}'_1 & \mathsf{r}'^{(\Omega)}_1 \\ \vdots & \vdots \\ \tilde{w}'_{\tilde{K}} & \mathsf{r}'^{(\Omega)}_{\tilde{K}} \\ \mathsf{m}_R & \mathsf{r}'_R \end{pmatrix} \tag{56}$$

If $\mathsf{L} = \tilde{K}+1$ and $\big(\vec{\epsilon}^{\,(1,\ell)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}\big)_{\ell\in[\mathsf{L}]}$ are chosen at uniformly random, then the probability that the following matrix E:

$$\mathsf{E} \triangleq \begin{pmatrix} \epsilon_1^{(1)} & \cdots & \epsilon_{\tilde{K}}^{(1)} & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_1^{(\mathsf{L})} & \cdots & \epsilon_{\tilde{K}}^{(\mathsf{L})} & 1 \end{pmatrix}$$

has zero determinant is negligible in the size of $\mathbb{Z}_q$ by an argument based on Schwartz–Zippel Lemma. Hence, $\big(\mathsf{m}_R, \mathsf{r}'_R, \mathsf{r}'_S, (\tilde{w}'_k, \mathsf{r}'^{(\Omega)}_k, \mathsf{r}'^{(T)}_k)_{k\in[\tilde{K}]}, (\mathsf{m}_j^{(S)})_{j\in[\tilde{J}]}, (\tau'_{j,k})_{j=1,k=1}^{\tilde{J}\ \tilde{K}}\big)$ can be uniquely determined by multiplying E^{-1} to both sides of Eqns. (55)-(56). Note that if we let $\big((\Omega_k, \tilde{T}_k)_{k\in[\tilde{K}]}, \tilde{R}, \tilde{S}\big)$ by the following

$$\Omega_k = G^{\tilde{w}'_k} \cdot Q^{\mathsf{r}'^{(\Omega)}_k}, \quad \tilde{T}_k = \prod_{j\in[\tilde{J}]} H_j^{\tau'_{j,k}} \cdot Q^{\mathsf{r}'^{(T)}_k},$$

$$\tilde{R} = G^{\mathsf{m}_R} \cdot Q^{\mathsf{r}'_R}, \quad \tilde{S} = \prod_{j\in[\tilde{J}]} H_j^{\mathsf{m}_j^{(S)}} \cdot Q^{\mathsf{r}'_S} \tag{57}$$

Then by Eqns. (55)-(56), one can check that $\big((\Omega_k, \tilde{T}_k)_{k\in[\tilde{K}]}, \tilde{R}, \tilde{S}\big)$ can satisfy Eqns. (54). Hence, $\big(\mathsf{m}_R, \mathsf{r}'_R, \mathsf{r}'_S, (\tilde{w}'_k, \mathsf{r}'^{(\Omega)}_k, \mathsf{r}'^{(T)}_k)_{k\in[\tilde{K}]}, (\mathsf{m}_j^{(S)})_{j\in[\tilde{J}]}, (\tau'_{j,k})_{j=1,k=1}^{\tilde{J}\ \tilde{K}}\big)$ is a satisfying witness.

(2) *Stage 2*: We first obtain additional random challenges $(\alpha^{(\rho)} \xleftarrow{\$} \mathbb{Z}_p^*)_{\rho=2}^{\mathsf{P}}$ by rewinding $\mathsf{P}-1$ times to Step (28). Then, we obtain different random challenges $(\vec{\epsilon}^{\,(\rho,1)} \xleftarrow{\$} \mathbb{Z}_p^*)_{\rho=2}^{\mathsf{P}}$, correspondingly. For each challenge tuple $(\alpha^{(\rho)}, \vec{\epsilon}^{\,(\rho,1)})$ where $\rho \in [\mathsf{P}]$, we obtain accepting transcript $\big((v_{j,k}^{(\rho,1)}, \mu_{j,k}^{(\rho,1)})_{(j,k)\in\mathcal{B}}, \tilde{\eta}_1^{(\rho,1)}, \tilde{\eta}_2^{(\rho,1)}, \tilde{\eta}_3^{(\rho,1)}\big)$ which the verifier checks at Step (32) to satisfy the following:

$$G^{\sum_{(j,k)\in\mathcal{B}} \mu'^{(\rho,1)}_{j,k}} \cdot Q^{\tilde{\eta}_2^{(\rho,1)}} \stackrel{?}{=} \prod_{c=0}^{B-1} M_c^{\frac{1}{\alpha^{(\rho)}+c}} \cdot \prod_{k\in[\tilde{K}]} V_k^{(\epsilon_k^{(\rho,1)})^{-1}} \tag{58}$$

We extract $\big((v'_k, \mathsf{r}'^{(V)}_k)_{k\in[\tilde{K}]}, (m'_c, \mathsf{r}'^{(M)}_c)_{c=0}^{B-1}\big)$ by

$$\begin{pmatrix} \sum_{(j,k)\in\mathcal{B}} \mu'^{(1,1)}_{j,k} & \tilde{\eta}_3^{(1,1)} \\ \vdots & \vdots \\ \sum_{(j,k)\in\mathcal{B}} \mu'^{(\mathsf{P},1)}_{j,k} & \tilde{\eta}_3^{(\mathsf{P},1)} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{\alpha^{(1)}} & \cdots & \frac{1}{\alpha^{(1)}+B-1} & (\epsilon_1^{(1,1)})^{-1} & \cdots & (\epsilon_{\tilde{K}}^{(1,1)})^{-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha^{(\mathsf{P})}} & \cdots & \frac{1}{\alpha^{(\mathsf{P})}+B-1} & (\epsilon_1^{(\mathsf{P},1)})^{-1} & \cdots & (\epsilon_{\tilde{K}}^{(\mathsf{P},1)})^{-1} \end{pmatrix} \cdot \begin{pmatrix} m'_0 & \mathsf{r}'^{(M)}_0 \\ \vdots & \vdots \\ m'_{B-1} & \mathsf{r}'^{(M)}_{B-1} \\ v'_1 & \mathsf{r}'^{(V)}_1 \\ \vdots & \vdots \\ v'_{\tilde{K}} & \mathsf{r}'^{(V)}_{\tilde{K}} \end{pmatrix} \tag{59}$$

If $\mathsf{P} = B+\tilde{K}$ and $\big(\alpha^{(\rho)} \xleftarrow{\$} \mathbb{Z}_p^*, \vec{\epsilon}^{\,(\rho,1)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}\big)_{\rho=1}^{\mathsf{P}}$ are chosen at uniformly random, then the probability that the following matrix A:

$$\mathsf{A} \triangleq \begin{pmatrix} \frac{1}{\alpha^{(1)}} & \cdots & \frac{1}{\alpha^{(1)}+B-1} & (\epsilon_1^{(1,1)})^{-1} & \cdots & (\epsilon_{\tilde{K}}^{(1,1)})^{-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha^{(\mathsf{P})}} & \cdots & \frac{1}{\alpha^{(\mathsf{P})}+B-1} & (\epsilon_1^{(\mathsf{P},1)})^{-1} & \cdots & (\epsilon_{\tilde{K}}^{(\mathsf{P},1)})^{-1} \end{pmatrix}$$

has zero determinant negligible in the size of $\mathbb{Z}_q$ by an argument based Schwartz–Zippel Lemma. Hence, $\big((v'_k, \mathsf{r}'^{(V)}_k)_{k\in[\tilde{K}]}, (m'_c, \mathsf{r}'^{(M)}_c)_{c=0}^{B-1}\big)$ can be uniquely determined by multiplying A^{-1} to both sides of Eqns. (59). Note that if we let $\big((V_k)_{k\in[\tilde{K}]}, (M_c)_{c=0}^{B-1}\big)$ by the following

$$M_c = G^{m'_c} \cdot Q^{\mathsf{r}'^{(M)}_c}, \qquad V_k = G^{v'_k} \cdot Q^{\mathsf{r}'^{(V)}_k} \tag{60}$$

then by Eqns. (55)-(56) $\big((V_k)_{k\in[\tilde{K}]}, (M_c)_{c=0}^{B-1}\big)$ can satisfy Eqn. (32). Namely, $\big(\mathsf{m}_V, \mathsf{r}'_R, (v'_k, \mathsf{r}'^{(V)}_k)_{k\in[\tilde{K}]}, (m'_c, \mathsf{r}'^{(M)}_c)_{c=0}^{B-1}\big)$ is a satisfying witness.

Together, we extracted $\big(\mathsf{m}_R, \mathsf{r}'_R, \mathsf{r}'_S, (\tilde{w}'_k, \mathsf{r}'^{(\Omega)}_k, \mathsf{r}'^{(T)}_k)_{k\in[\tilde{K}]}, (\mathsf{m}^{(S)}_j)_{j\in[\tilde{J}]}, (\tau'_{j,k})^{\tilde{J}}_{j=1,k=1}, \mathsf{m}_V, \mathsf{r}'_R, (v'_k, \mathsf{r}'^{(V)}_k)_{k\in[\tilde{K}]}, (m'_c, \mathsf{r}'^{(M)}_c)^{B-1}_{c=0}\big)$ as a satisfying witness.

Next, consider $\alpha^{(1)}$ and pick $\vec{\epsilon}^{\,(1,1)}$ and $\vec{\epsilon}^{\,(1,2)}$ from $\{\vec{\epsilon}^{\,(\rho,\ell)}\}_{\ell\in[\mathsf{L}]}$. One can extract witness $(\tilde{w}'_{j,k}, \mathsf{r}'^{(\nu)}_{j,k}, f'_{j,k}, \mathsf{r}'^{(\mu)}_{j,k})$ by solving

$$\nu^{(1,1)}_{j,k} \triangleq \tilde{w}'_{j,k}\cdot\epsilon^{(1,1)}_k + \mathsf{r}'^{(\nu)}_{j,k}, \qquad \nu^{(1,2)}_{j,k} \triangleq \tilde{w}'_{j,k}\cdot\epsilon^{(1,2)}_k + \mathsf{r}'^{(\nu)}_{j,k}$$

$$\mu^{(1,1)}_{j,k} \triangleq \hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot\epsilon^{(1,1)}_k + \mathsf{r}'^{(\mu)}_{j,k}, \qquad \mu^{(1,2)}_{j,k} \triangleq \hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot\epsilon^{(1,2)}_k + \mathsf{r}'^{(\mu)}_{j,k}$$

We substitute $(\tilde{w}'_{j,k}, \mathsf{r}'^{(\nu)}_{j,k}, f'_{j,k}, \mathsf{r}'^{(\mu)}_{j,k})_{j\in[\tilde{J}],k\in[\tilde{K}]}$ into $(\tilde{\nu}^{(1,1)}_{j,k}, \tilde{\mu}^{(1,1)}_{j,k})_{j\in[\tilde{J}],k\in[\tilde{K}]}$ to obtain

$$\begin{aligned}
&\prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]}\tilde{\nu}^{(1,1)}_{j,k}\cdot\tilde{\mu}^{(1,1)}_{j,k}}\cdot Q^{\tilde{\eta}^{(1,1)}_1}\\
&= \prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}}\hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot(\alpha^{(1)}\cdot\hat{B}_{j,k}+\tilde{w}'_{j,k})\cdot(\epsilon^{(1,1)}_k)^2}\cdot\prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}}(\epsilon^{(1,1)}_k)^2}\\
&\cdot\prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}}(\hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot\mathsf{r}'^{(j)}_{\nu,k}+(\alpha^{(1)}\cdot\hat{B}_{j,k}+\tilde{w}'_{j,k})\cdot\mathsf{r}'^{(j)}_{\mu,k})\cdot\epsilon^{(1,1)}_k+\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}}\mathsf{r}'^{(\mu)}_{j,k}\cdot\mathsf{r}'^{(\nu)}_{j,k}}\\
&\cdot Q^{\tilde{\eta}^{(1,1)}_1}
\end{aligned} \tag{61}$$

We then compare Eqn. (61) with Eqn. (54) (setting $\ell = 1$) and Eqn. (57). Based on the DLR assumption, we obtain the exponent of each H_j as

$$\begin{aligned}
&\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}}\hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot(\alpha^{(1)}\cdot\hat{B}_{j,k}+\tilde{w}'_{j,k})\cdot(\epsilon^{(1,1)}_k)^2\\
&+\sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}}(\epsilon^{(1,1)}_k)^2\\
&+\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}}(\hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot\mathsf{r}'^{(j)}_{\nu,k}+(\alpha^{(1)}\cdot\hat{B}_{j,k}+\tilde{w}'_{j,k})\cdot\mathsf{r}'^{(j)}_{\mu,k})\cdot\epsilon^{(1,1)}_k\\
&+\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}}\mathsf{r}'^{(\mu)}_{j,k}\cdot\mathsf{r}'^{(\nu)}_{j,k}\\
&=\sum_{k\in[\tilde{K}]}(\epsilon^{(1,1)}_k)^2+\sum_{k\in[\tilde{K}]}\tau'_{j,k}\cdot\epsilon^{(1,1)}_k+\mathsf{m}^{(S)}_j
\end{aligned} \tag{62}$$

Note that $\vec{\epsilon}^{\,(1,1)}$ is selected at uniformly random. Hence, Eqn. (62) is always true with high probability, only if the coefficients of $(\epsilon^{(1,1)}_k)^2$ and $\epsilon^{(1,1)}_k$ for all $k\in[\tilde{K}]$ and constant terms of LHS and RHS are equivalent. Particular, we consider the coefficient of $(\epsilon^{(1,1)}_k)^2$ for each $(j,k)\in\mathcal{B}$, and obtain

$$\hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot(\alpha^{(1)}\cdot\hat{B}_{j,k}+\tilde{w}'_{j,k}) = 1$$

Similarly, we substitute $(f'_{j,k}, \mathsf{r}'^{(\mu)}_{j,k})_{(j,k)\in\mathcal{B}}$ into $(\mu'^{(1,1)}_{j,k})_{(j,k)\in\mathcal{B}}$ to obtain

$$G^{\sum_{(j,k)\in\mathcal{B}}\mu'^{(1,1)}_{j,k}}\cdot Q^{\tilde{\eta}^{(1,1)}_2} = G^{\sum_{(j,k)\in\mathcal{B}}f'_{j,k}+\hat{B}_{j,k}\cdot\mathsf{r}'^{(\mu)}_{j,k}\cdot\epsilon^{-1}_k}\cdot Q^{\tilde{\eta}^{(1,1)}_2} \tag{63}$$

We compare Eqn. (63) with Eqn. (58) (setting $\rho = 1$) and Eqn. (60). Based on the DLR assumption, and comparison with the constant terms (that are independent of ϵ^{-1}_k), we obtain

$$\sum_{(j,k)\in\mathcal{B}} f'_{j,k} = \sum_{c=0}^{B-1}\frac{m'_c}{\alpha^{(1)}+c}$$

Overall, we obtain

$$\begin{cases}\hat{B}^{-1}_{j,k}\cdot f'_{j,k}\cdot(\alpha^{(1)}\cdot\hat{B}_{j,k}+\tilde{w}_{j,k}) = 1, & \forall(j,k)\in\mathcal{B}\\ \sum_{(j,k)\in\mathcal{B}} f'_{j,k} = \sum_{c=0}^{B-1}\frac{m'_c}{\alpha^{(1)}+c}\end{cases} \tag{64}$$

Hence, this implies $\tilde{w}_{j,k}\in\{0,\hat{B}_{j,k},...,(B-1)\cdot\hat{B}_{j,k}\}$.

Finally, we substitute $(\tilde{w}'_{j,k}, \mathsf{r}'_{j,k})_{j\in[\tilde{J}],k\in[\tilde{K}]}$ into $(\nu^{(1,1)}_{j,k})_{j\in[\tilde{J}],k\in[\tilde{K}]}$ to obtain

$$\sum_{j\in[\tilde{J}]}\tilde{w}'_{j,k} = \tilde{w}'_k$$

As in type-1 argument, we obtain

$$G^{\omega}\cdot Q^{\mathsf{r}_\omega} = \mathtt{Cm}(\omega) = \prod_{k\in[\tilde{K}]}\Omega_k = G^{\sum_{k\in[\tilde{K}]}\sum_{j\in[\tilde{J}]}\tilde{w}'_{j,k}}\cdot Q^{\sum_{k\in[\tilde{K}]}\mathsf{r}'^{(\Omega)}_k}$$

By the DLR assumption, we obtain $\omega = \sum_{k\in[\tilde{K}]}\sum_{j\in[\tilde{J}]}\tilde{w}'_{j,k}$. Therefore, this proves that the extracted $(\tilde{w}_{j,k}\in\{0,\hat{B}_{j,k},...,(B-1)\cdot\hat{B}_{j,k}\})_{(j,k)\in\mathcal{B}}$ is a valid witness to $\omega\in[0,B^{\tilde{N}}]$.

□

E Proofs for VeRange Type-3 Range Argument

Lemma E.1. *BCCGP polynomial commitment scheme satisfies perfect completeness and $(V+1)$-special soundness.*

See the proofs of perfect completeness and special soundness in [4]. We need the following slightly stronger SHVZK lemma of BCCGP polynomial commitment scheme.

Lemma E.2. *Given a fixed output value y_{F}, BCCGP polynomial commitment scheme satisfies SHVZK.*

Proof. We define a simulator as follows. Given y_{F}, the simulator first generates part of the commitment $(H_u)^U_{u=1}$ and part of the proof $(\mathsf{f}_v)^V_{v=1}$ at uniformly random. Next, after learning the challenge x from the verifier, the simulator rewinds to set

$$\mathsf{f}_0 \triangleq y_{\mathsf{F}} - \sum_{v=1}^{V}\mathsf{f}_v\cdot x^{(v-1)U+\xi}$$

and then set

$$H_0 \triangleq \frac{\prod_{v=0}^{V}G_v^{\mathsf{f}_v}}{\prod_{u=1}^{U}H_u^{x^u}}$$

One can check that the above settings of $\mathtt{Cm}_{\mathsf{F}} = (H_u)^U_{u=0}$ and $\pi_{\mathsf{F}} = (\mathsf{f}_v)^V_{v=0}$ can successfully pass the verification at the verifier without the witness F. Moreover, the transcripts appear to be uniformly random. □

Theorem E.3. VeRange *type-3 range argument protocol Π_{ty3} satisfies perfect completeness, SHVZK and CWE.*

Proof. **Perfect Completeness**: Π_{ty3} satisfies perfect completeness by following the perfect completeness of BCCGP polynomial commitment scheme and Eqns. (37)-(38).

SHVZK: We define a simulator as follows. First, by SHVZK of BCCGP polynomial commitment scheme in Lemma E.2, given any y_{B} and y_{S} the simulator can generate valid $(\mathtt{Cm}_{\mathsf{B}}, \pi_{\mathsf{B}})$ and $(\mathtt{Cm}_{\mathsf{S}}, \pi_{\mathsf{S}})$ without knowing the true witnesses $\mathsf{B}[X]$ and $\mathsf{S}[X]$ to any challenge x. Then the simulator generates group elements $\big((D_k)_{k\in[\tilde{K}]}, (\Omega_k)_{k\in[\tilde{K}-1]},\big)$ at Steps (39)-(40) and field elements $\big((\bar{d}_j)_{j\in[\tilde{J}]}, \tilde{\eta}_1, \tilde{\eta}_2, y_{\mathsf{S}}\big)$ at Steps (41)-(43) at uniformly random. Then the simulator sets

$$\Omega_{\tilde{K}} \triangleq \mathtt{Cm}(\omega) \cdot \Big(\prod_{k\in[\tilde{K}-1]} \Omega_k\Big)^{-1}$$

Next, after learning the challenges (β, x) from the verifier, the simulator rewinds to Steps (39),(40),(41) to set

$$\begin{cases} R_1 \triangleq \Big(\prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j} \cdot Q^{\tilde{\eta}_1} \cdot \prod_{k\in[\tilde{K}]} (D_k)^{-\mathsf{L}_k[x]}\Big)^{-\mathsf{L}_0[x]} \\ y_{\mathsf{B}} \triangleq \frac{1}{\mathsf{L}_0[x]} \cdot \Big(\sum_{j\in[\tilde{J}]} \beta^j \cdot \bar{d}_j \cdot (\bar{d}_j - 1)\cdots(\bar{d}_j - B + 1)\Big) \\ R_2 \triangleq \Big(G^{(y_{\mathsf{S}}\cdot\mathsf{L}_0[x] + \sum_{j\in[\tilde{J}]} \bar{d}_j\cdot\bar{B}_j)} \cdot Q^{\tilde{\eta}_2} \cdot \prod_{k\in[\tilde{K}]} (\Omega_k)^{-\mathsf{L}_k[x]}\Big)^{-\mathsf{L}_0[x]} \end{cases}$$

One can check that the above settings of $\big((D_k)_{k\in[\tilde{K}]}, (\Omega_k)_{k\in[\tilde{K}]}, R_1, R_2, \mathtt{Cm}_{\mathsf{B}}, \mathtt{Cm}_{\mathsf{S}}\big)$ and $\big((v_{j,k}, \mu_{j,k})_{(j,k)\in\mathcal{B}}, \tilde{\eta}_1, \tilde{\eta}_2, \tilde{\eta}_3, y_{\mathsf{B}}, y_{\mathsf{S}}, \pi_{\mathsf{B}}, \pi_{\mathsf{S}}\big)$ can successfully pass the verification at the verifier without the witness ω. Moreover, the transcripts appear to be uniformly random.

CWE: We define an emulator as follows. Given the initial message $\big((\Omega_k, D_k)_{k\in[\tilde{K}]}, R_1, R_2, \mathtt{Cm}_{\mathsf{S}}\big)$ from the prover in an honest execution of Π_{ty3}, we proceed to rewinding. We generate $\mathsf{P}\times\mathsf{L}$ challenge tuples $\{(\beta^{(\rho)}, x^{(\ell)})\}_{\rho\in[\mathsf{P}],\ell\in[\mathsf{L}]}$. For each challenge tuple, we obtain the corresponding transcript $\big(\mathtt{Cm}_{\mathsf{B}}^{(\rho)}, (\bar{d}_j^{(\rho,\ell)})_{j\in\tilde{J}}, y_{\mathsf{B}}^{(\rho,\ell)}, \pi_{\mathsf{B}}^{(\rho,\ell)}, y_{\mathsf{S}}^{(\rho,\ell)}, \pi_{\mathsf{S}}^{(\rho,\ell)}, \tilde{\eta}_1^{(\rho,\ell)}, \tilde{\eta}_2^{(\rho,\ell)}\big)$ which the verifier checks at Step (44) to satisfy the following:

$$\begin{cases} \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_{\mathsf{B}}^{(\rho)}, x^{(\ell)}, y_{\mathsf{B}}^{(\rho,\ell)}, \pi_{\mathsf{B}}^{(\rho,\ell)}] \stackrel{?}{=} 1 \\ \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_{\mathsf{S}}, x^{(\ell)}, y_{\mathsf{S}}^{(\rho,\ell)}, \pi_{\mathsf{S}}^{(\rho,\ell)}] \stackrel{?}{=} 1 \\ \prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j^{(\rho,\ell)}} \cdot Q^{\tilde{\eta}_1^{(\rho,\ell)}} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (D_k)^{\mathsf{L}_k[x^{(\ell)}]} \cdot R_1^{\mathsf{L}_0[x^{(\ell)}]} \\ \sum_{j\in[\tilde{J}]} (\beta^{(\rho)})^j \cdot \bar{d}_j^{(\rho,\ell)} \cdot (\bar{d}_j^{(\rho,\ell)} - 1)\cdots(\bar{d}_j^{(\rho,\ell)} - B + 1) \stackrel{?}{=} y_{\mathsf{B}}^{(\rho,\ell)} \cdot \mathsf{L}_0[x^{(\ell)}] \\ G^{(y_{\mathsf{S}}^{(\rho,\ell)}\cdot\mathsf{L}_0[x^{(\ell)}] + \sum_{j\in[\tilde{J}]} \bar{d}_j^{(\rho,\ell)}\cdot\bar{B}_j)} \cdot Q^{\tilde{\eta}_2^{(\rho,\ell)}} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (\Omega_k)^{\mathsf{L}_k[x^{(\ell)}]} \cdot R_2^{\mathsf{L}_0[x^{(\ell)}]} \\ \mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \end{cases} \tag{65}$$

Next, considering $\big(\beta^{(1)}, (x^{(\ell)})_{\ell\in\tilde{J}}\big)$, we extract $(\hat{d}'_{j,k})_{j\in[\tilde{J}],k\in[\tilde{K}]}$, such that

$$\begin{pmatrix} \bar{d}_1^{(1,1)} & \cdots & \bar{d}_{\tilde{J}}^{(1,1)} & \tilde{\eta}_1^{(1,1)} \\ \vdots & \ddots & \vdots & \vdots \\ \bar{d}_1^{(1,\mathsf{L})} & \cdots & \bar{d}_{\tilde{J}}^{(1,\mathsf{L})} & \tilde{\eta}_1^{(1,\mathsf{L})} \end{pmatrix}$$

$$= \begin{pmatrix} \mathsf{L}_1[x^{(1)}] & \cdots & \mathsf{L}_{\tilde{K}}[x^{(1)}] & \mathsf{L}_{\tilde{0}}[x^{(1)}] \\ \vdots & \ddots & \vdots & \vdots \\ \mathsf{L}_1[x^{(\mathsf{L})}] & \cdots & \mathsf{L}_{\tilde{K}}[x^{(\mathsf{L})}] & \mathsf{L}_{\tilde{0}}[x^{(\mathsf{L})}] \end{pmatrix} \cdot \begin{pmatrix} \hat{d}'_{1,1} & \cdots & \hat{d}'_{\tilde{J},1} & \mathsf{r}_1'^{(\mathsf{D})} \\ \vdots & \ddots & \vdots & \vdots \\ \hat{d}'_{1,\tilde{K}} & \cdots & \hat{d}'_{\tilde{J},\tilde{K}} & \mathsf{r}_{\tilde{K}}'^{(\mathsf{D})} \\ \mathsf{r}_1'^{(\mathsf{d})} & \cdots & \mathsf{r}_{\tilde{J}}'^{(\mathsf{d})} & \mathsf{r}_1' \end{pmatrix} \tag{66}$$

by noting that the following matrix L is invertible with high probability based on Schwartz–Zippel Lemma, when $\mathsf{L} = \tilde{K} + 1$:

$$\mathsf{L} \triangleq \begin{pmatrix} \mathsf{L}_1[x^{(1)}] & \cdots & \mathsf{L}_{\tilde{K}}[x^{(1)}] & \mathsf{L}_{\tilde{0}}[x^{(1)}] \\ \vdots & \ddots & \vdots & \vdots \\ \mathsf{L}_1[x^{(\mathsf{L})}] & \cdots & \mathsf{L}_{\tilde{K}}[x^{(\mathsf{L})}] & \mathsf{L}_{\tilde{0}}[x^{(\mathsf{L})}] \end{pmatrix}$$

Note that if $\bar{d}_j^{(1,1)} \neq \bar{d}_j^{(\rho,1)}$ for some ρ, j, then we would obtain a non-trivial logarithmic relation:

$$\prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j^{(1,1)}} \cdot Q^{\tilde{\eta}_1^{(1,1)}} = \prod_{k\in[\tilde{K}]} (D_k)^{\mathsf{L}_k[x^{(1)}]} \cdot R_1^{\mathsf{L}_0[x^{(1)}]} = \prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j^{(\rho,1)}} \cdot Q^{\tilde{\eta}_1^{(\rho,1)}}$$

By the DLR assumption, we conclude that $\bar{d}_j^{(1,\ell)} = \bar{d}_j^{(\rho,\ell)}$ for all ρ, j. In the following, we write $\bar{d}_j^{(\ell)} = \bar{d}_j^{(\rho,\ell)}$, by dropping the dependence on ρ.

We can extract $\mathsf{B}^{(\rho)}[X]$ from $\mathtt{Cm}_{\mathsf{B}}^{(\rho)}$ because of the special soundness of BCCGP polynomial commitment. Note that the extracted polynomial $\mathsf{B}^{(\rho)}[X]$ does not depend on $x^{(\ell)}$. We can re-express $\mathsf{B}^{(\rho)}[X]$ by $\mathsf{B}^{(\rho)}[x^{(\ell)}] = \sum_{j\in[\tilde{J}]} (\beta^{(\rho)})^j \cdot \mathsf{B}_j[x^{(\ell)}]$ for inputs $\{x^{(\ell)}\}_{\ell\in[\mathsf{L}]}$. Recall that $y_{\mathsf{B}}^{(\rho,\ell)} \triangleq \mathsf{B}^{(\rho)}[x^{(\ell)}]$. We also define $y_{\mathsf{B}_j}^{(\ell)} \triangleq \mathsf{B}_j[x^{(\ell)}]$, which can be extracted from

$$\begin{pmatrix} y_{\mathsf{B}}^{(1,1)} & \cdots & y_{\mathsf{B}}^{(1,\mathsf{L})} \\ \vdots & \ddots & \vdots \\ y_{\mathsf{B}}^{(\mathsf{P},1)} & \cdots & y_{\mathsf{B}}^{(\mathsf{P},\mathsf{L})} \end{pmatrix} = \begin{pmatrix} \beta^{(1)} & \cdots & (\beta^{(1)})^{\tilde{J}} \\ \vdots & \ddots & \vdots \\ \beta^{(\mathsf{P})} & \cdots & (\beta^{(\mathsf{P})})^{\tilde{J}} \end{pmatrix} \cdot \begin{pmatrix} y_{\mathsf{B}_1}^{(1)} & \cdots & y_{\mathsf{B}_1}^{(\mathsf{L})} \\ \vdots & \ddots & \vdots \\ y_{\mathsf{B}_{\tilde{J}}}^{(1)} & \cdots & y_{\mathsf{B}_{\tilde{J}}}^{(\mathsf{L})} \end{pmatrix} \tag{67}$$

$(y_{\mathsf{B}_j}^{(\ell)})_{\ell\in[\mathsf{L}]}$ can uniquely determined, when the following Vandermonde matrix B is invertible with high probability, $\mathsf{P} = \tilde{J}$

$$\mathsf{B} \triangleq \begin{pmatrix} \beta^{(1)} & \cdots & (\beta^{(1)})^{\tilde{J}} \\ \vdots & \ddots & \vdots \\ \beta^{(\mathsf{P})} & \cdots & (\beta^{(\mathsf{P})})^{\tilde{J}} \end{pmatrix}$$

Note the degree of $\mathsf{B}[X]$ is no more than $B\tilde{K}$. If $\mathsf{L} > B\tilde{K}$, then interpolating $\{(x^{(\ell)}, y_{\mathsf{B}_j}^{(\ell)})\}_{\ell\in[\mathsf{L}]}$ can uniquely determine $\mathsf{B}_j[X]$.

Next, by the fourth equation in Eqn. (65), we obtain

$$\sum_{j\in[\tilde{J}]} (\beta^{(\rho)})^j \cdot \bar{d}_j^{(\ell)} \cdot (\bar{d}_j^{(\ell)} - 1)\cdots(\bar{d}_j^{(\ell)} - B + 1)$$
$$= y_{\mathsf{B}}^{(\rho,\ell)} \cdot \mathsf{L}_0[x^{(\ell)}]$$

$$= \Big(\sum_{j\in[\tilde{J}]} (\beta^{(\rho)})^j \cdot \mathsf{B}_j[x^{(\ell)}] \Big) \cdot \mathsf{L}_0[x^{(\ell)}] \tag{68}$$

Since $\beta^{(\rho)} \xleftarrow{\$} \mathbb{Z}_p$ is selected at random, by comparing the coefficients of $(\beta^{(\rho)})^j$, we obtain

$$\bar{d}_j^{(\ell)} \cdot (\bar{d}_j^{(\ell)} - 1) \cdots (\bar{d}_j^{(\ell)} - B + 1) = \mathsf{B}_j[x^{(\ell)}] \cdot \mathsf{L}_0[x^{(\ell)}]$$

for $j \in [\tilde{J}]$. Since $\bar{d}_j^{(\ell)} = \mathsf{r}_j'^{(\mathsf{d})} \cdot \mathsf{L}_0[x^{(\ell)}] + \sum_{k\in[\tilde{K}]} \hat{d}'_{j,k} \cdot \mathsf{L}_k[x^{(\ell)}]$. Note that $\mathsf{L} > B\tilde{K}$. Hence, interpolating $\{(x^{(\ell)})\}_{\ell\in[\mathsf{L}]}$ can uniquely determine the whole polynomial. Therefore, we obtain

$$\Big(\mathsf{r}_j'^{(\mathsf{d})} \cdot \mathsf{L}_0[X] + \sum_{k\in[\tilde{K}]} \hat{d}'_{j,k} \cdot \mathsf{L}_k[X]\Big) \cdot \Big(\mathsf{r}_j'^{(\mathsf{d})} \cdot \mathsf{L}_0[X] + \sum_{k\in[\tilde{K}]} \hat{d}'_{j,k} \cdot \mathsf{L}_k[X] - 1\Big)$$
$$\cdots \Big(\mathsf{r}_j'^{(\mathsf{d})} \cdot \mathsf{L}_0[X] + \sum_{k\in[\tilde{K}]} \hat{d}'_{j,k} \cdot \mathsf{L}_k[X] - B + 1\Big) = \mathsf{B}_j[X] \cdot \mathsf{L}_0[X]$$

By substituting $X \in \{z_1, ..., z_{\tilde{K}}\}$, we obtain

$$\hat{d}'_{j,k} \cdot (\hat{d}'_{j,k} - 1) \cdots (\hat{d}'_{j,k} - B + 1) = 0$$

for $j \in [\tilde{J}], k \in [\tilde{K}]$.

Similarly, we can extract $\mathsf{S}[X]$ from $\mathtt{Cm}_\mathsf{S}$ because of the special soundness of BCCGP polynomial commitment. Next, we extract $\big(\mathsf{s}', \mathsf{r}'_2, (\tilde{w}'_k, \mathsf{r}'^{(\Omega)}_k)_{k\in[\tilde{K}]}\big)$ by

$$\begin{pmatrix} y_\mathsf{S}^{(1,1)} \cdot \mathsf{L}_0[x^{(1)}] + \sum_{j\in[\tilde{J}]} \bar{d}_j^{(1)} \cdot \bar{B}_j & \bar{\eta}_2^{(1,1)} \\ \vdots & \vdots \\ y_\mathsf{S}^{(1,\mathsf{L})} \cdot \mathsf{L}_0[x^{(\mathsf{L})}] + \sum_{j\in[\tilde{J}]} \bar{d}_j^{(\mathsf{L})} \cdot \bar{B}_j & \bar{\eta}_2^{(1,\mathsf{L})} \end{pmatrix}$$
$$= \begin{pmatrix} \mathsf{L}_1[x^{(1)}] & \cdots & \mathsf{L}_{\tilde{K}}[x^{(1)}] & \mathsf{L}_{\tilde{0}}[x^{(1)}] \\ \vdots & \ddots & \vdots & \vdots \\ \mathsf{L}_1[x^{(\mathsf{L})}] & \cdots & \mathsf{L}_{\tilde{K}}[x^{(\mathsf{L})}] & \mathsf{L}_{\tilde{0}}[x^{(\mathsf{L})}] \end{pmatrix} \cdot \begin{pmatrix} \tilde{w}'_1 & \mathsf{r}'^{(\Omega)}_1 \\ \vdots & \vdots \\ \tilde{w}'_{\tilde{K}} & \mathsf{r}'^{(\Omega)}_{\tilde{K}} \\ \mathsf{s}' & \mathsf{r}'_2 \end{pmatrix} \tag{69}$$

By interpolating $\{(x^{(\ell)})\}_{\ell\in[\mathsf{L}]}$ can uniquely determine the whole polynomial. Therefore, we obtain

$$\mathsf{S}[X] \cdot \mathsf{L}_0[X] + \sum_{j\in[\tilde{J}]} \Big(\sum_{k\in[\tilde{K}]} \hat{d}'_{j,k} \cdot \mathsf{L}_k[X] \Big) \cdot \Big(\sum_{k\in[\tilde{K}]} \hat{B}_{j,k} \cdot \mathsf{L}_k[X] \Big)$$
$$= \sum_{k\in[\tilde{K}]} \tilde{w}'_k \cdot \mathsf{L}_k[X] + \mathsf{s}' \cdot \mathsf{L}_0[X] \tag{70}$$

By substituting $X \in \{z_1, ..., z_{\tilde{K}}\}$, we obtain $\sum_{j\in[\tilde{J}]} \hat{d}'_{j,k} \cdot \hat{B}_{j,k} = \tilde{w}'_k$ for $k \in [\tilde{K}]$.

Finally, as in type-1 argument, we obtain

$$G^\omega \cdot Q^{\mathsf{r}_\omega} = \mathtt{Cm}(\omega) = \prod_{k\in[\tilde{K}]} \Omega_k = G^{\sum_{k\in[\tilde{K}]} \sum_{j\in[\tilde{J}]} \hat{d}'_{j,k} \cdot \hat{B}_{j,k}} \cdot Q^{\sum_{k\in[\tilde{K}]} \mathsf{r}'^{(\Omega)}_k}$$

By the DLR assumption, we obtain $\omega = \sum_{k\in[\tilde{K}]} \sum_{j\in[\tilde{J}]} \hat{d}'_{j,k} \cdot \hat{B}_{j,k}$. Therefore, this proves that the extracted $(\tilde{w}_{j,k} \in \{0, \hat{B}_{j,k}, ..., (B-1) \cdot \hat{B}_{j,k}\})_{(j,k)\in\mathcal{B}}$ is a valid witness to $\omega \in [0, B^{\tilde{N}}]$. □

Figure 9: Flashproofs range argument protocol

$\Pi_{\text{flash}}\big[\mathtt{Cm}(\omega) \in \mathbb{G};\ \omega \in \mathbb{Z}_p, \mathsf{r}_\omega \in \mathbb{Z}_p^*\big]$

$\mathcal{P}: \vec{\mathbf{b}} \in \{0,1\}^N$ is the bit-decomposition of ω such that $\omega = \sum_{i\in[N]} b_i \cdot 2^{i-1}$

$\vec{\mathsf{r}} \xleftarrow{\$} \mathbb{Z}_p^{*J}$, $\vec{\mathsf{r}}^{(W)}, \vec{\mathsf{r}}^{(T)} \xleftarrow{\$} \mathbb{Z}_p^{*K}$, $\vec{\mathsf{r}}^{(\hat{T})} \xleftarrow{\$} \mathbb{Z}_p^{*K(K\text{-}1)/2}$, $\mathsf{r}_R, \mathsf{r}_S \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathsf{r}_K^{(W)} \triangleq \mathsf{r}_\omega - \sum_{k\in[K-1]} \mathsf{r}_k^{(W)}$

$\Big(w_{j,k} \triangleq \hat{b}_{j,k} \cdot \hat{2}_{j,k}\Big)_{j\in[J],k\in[K]}$, $\Big(t_{j,k} \triangleq \mathsf{r}_j \cdot (\hat{2}_{j,k} - 2w_{j,k})\Big)_{j\in[J],k\in[K]}$

$\Big(\hat{t}^{(j)}_{k,k'} \triangleq \sum_{j\in[J]} w_{j,k'}(\hat{2}_{j,k} - w_{j,k}) + w_{j,k}(\hat{2}_{j,k'} - w_{j,k'})\Big)_{k\in[K],k'\in[K]\setminus\{k\}}$

$\mathcal{P} \Rightarrow \mathcal{V}: \Big(W_k \triangleq G^{\sum_{j\in[J]} w_{j,k}} \cdot Q^{\mathsf{r}_k^{(W)}}\Big)_{k\in[K]}$, $\Big(T_k \triangleq \prod_{j\in[J]} H_j^{t_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\Big)_{k\in[K]}$

$\Big(\hat{T}_{k,k'} \triangleq \prod_{j\in[J]} H_j^{\hat{t}^{(j)}_{k,k'}} \cdot Q^{\mathsf{r}^{(\hat{T})}_{k,k'}}\Big)_{k\in[K],k'\in[K]\setminus\{k\}}$

$R \triangleq G^{\sum_{j\in[J]} \mathsf{r}_j} \cdot Q^{\mathsf{r}_R} \in \mathbb{G}, \qquad S \triangleq \prod_{j\in[J]} H_j^{-(\mathsf{r}_j)^2} \cdot Q^{\mathsf{r}_S}$

$\mathcal{P} \Leftarrow \mathcal{V}: \vec{\epsilon} \xleftarrow{\$} \mathbb{Z}_p^{*K}$

$\mathcal{P} \Rightarrow \mathcal{V}: \Big(v_j \triangleq \sum_{k\in[K]} w_{j,k} \cdot \epsilon_k + \mathsf{r}_j\Big)_{j\in[J]}, \quad \eta_2 \triangleq \vec{\mathsf{r}}^{(W)} \cdot \vec{\epsilon} + \mathsf{r}_R$

$\eta_1 \triangleq \sum_{k\in[K],k'\in[K]\setminus\{k\}} \mathsf{r}^{(\hat{T})}_{k,k'} \epsilon_k \epsilon_{k'} + \vec{\mathsf{r}}^{(T)} \cdot \vec{\epsilon} + \mathsf{r}_S$

$\mathcal{V}: \Big(u_j \triangleq \sum_{k\in[K]} \hat{2}_{j,k} \cdot \epsilon_k - v_j\Big)_{j\in[J]}$

$\textsc{Check}$
$\prod_{j\in[J]} H_j^{v_j \cdot u_j} \cdot Q^{\eta_1} \stackrel{?}{=} \prod_{k\in[K],k'\in[K]\setminus\{k\}} \hat{T}_{k,k'}^{\epsilon_k \epsilon_{k'}} \cdot \prod_{k\in[K]} T_k^{\epsilon_k} \cdot S$
$G^{\sum_{j\in[J]} v_j} \cdot Q^{\eta_2} \stackrel{?}{=} \prod_{k\in[K]} W_k^{\epsilon_k} \cdot R$
$\mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[K]} W_k$

F Flashproofs Range Argument

We include the Flashproofs range argument for completeness. Flashproofs range argument [30] utilizes a bit-decomposition approach for proving a number in a range in a similar fashion of VeRange type-1. But Flashproofs rely on a different aggregation technique. The Flashproofs range argument protocol is described in Fig. 9. The proof size includes $\frac{K^2+3K}{2}+2$ group elements and $J+2$ field elements. The verification time takes $J + \frac{K^2+3K}{2} + 3$ group exponentiations. The proving time takes $\frac{K(K-1)(J+1)}{2} + K(J+1) + 2K + J + 3$ group exponentiations. To minimize group exponentiations in verification, we set $J \approx N^{2/3}$ and $K \approx N^{1/3}$.

G More on VeRange Type-2 Range Argument

G.1 Type-2B Range Argument

While VeRange type-2 has only $O((\frac{N}{\log N})^{1/2})$ group exponentiations, there are considerably more field operations, which incur additional gas cost in practice. Therefore, we develop VeRange type-2B by combining the ideas of Flashproofs in Appendix F and the reciprocal relation from Bulletproofs++, which yields a lower gas cost alternative at the expense of $O((\frac{N}{\log N})^{2/3})$ group exponentiations in verification. Note that in Sec. 6, we observe that the empirical gas cost of VeRange type-2B is considerably lower than type-2. The full protocol of VeRange type-2B range argument is

described in Fig. 10. The perfect completeness, SHVZK and CWE of the type-2B range argument can be proved in a similar manner as the type-2 range argument.

Remarks: The proof size of VeRange type-2B range argument includes $3\tilde{K} + B + 3 + \frac{\tilde{K}(\tilde{K}-1)}{2}$ group elements and $2\tilde{J} + 4$ field elements. The verification takes $\tilde{J} + 2\tilde{K} + \frac{\tilde{K}(\tilde{K}-1)}{2} + B + 7$ group exponentiations. The proving takes $\tilde{J} + 5\tilde{K} + \tilde{K}^2 + 2B + 5$ group exponentiations. To minimize the number of group exponentiations in verification, we set $B \approx (\frac{N}{\log N})^{2/3}$ and $\tilde{J} \approx \left\lceil (\frac{3N}{2\log N})^{2/3} \right\rceil$ and $\tilde{K} \approx \left\lceil (\frac{3N}{2\log N})^{1/3} \right\rceil$. Hence, the verification takes around $(1 + (\frac{3}{2})^{5/2})(\frac{N}{\log N})^{2/3} + (\frac{3}{2})^{4/3}(\frac{N}{\log N})^{1/3}$ $\mathbb{G}$ group exponentiations and proving takes around $(2+18^{2/3})(\frac{N}{\log N})^{2/3} + \frac{5}{2}(\frac{3}{2})^{1/3}(\frac{N}{\log N})^{1/3}$ $\mathbb{G}$. Then, the proof size includes around $(1+(\frac{9}{32})^{1/3})(\frac{N}{\log N})^{2/3} + \frac{5}{2}(\frac{3}{2})^{1/3}(\frac{N}{\log N})^{1/3}$ group elements and around $2.6(\frac{N}{\log N})^{2/3} + 4$ field elements.

We can also construct the aggregated type-2B range argument protocol in a similar manner as the aggregated type-2 range argument protocol in Appendix G.2. We skip the full description of the aggregated type-2B range argument protocol because of page limit.

Figure 10: VeRange **type-2B range argument protocol**

$\Pi_{\text{ty2b}}\left[\mathtt{Cm}(\omega) \in \mathbb{G};\ \omega \in \mathbb{Z}_p, \mathsf{r}_\omega \in \mathbb{Z}_p^*\right]$

$\textsc{Setup}: \hat{B}_k \triangleq \sum_{j\in[\tilde{J}]:(j,k)\in\mathcal{B}} \hat{B}_{j,k}, \quad H_{\tilde{J}} \triangleq \prod_{j\in[\tilde{J}]} H_j$

$\mathcal{P}: \vec{\mathbf{d}} \in (\{0,...,B-1\})^{\tilde{N}}$ is the B-ary digit decompo. of ω such that $\omega = \sum_{i\in[\tilde{N}]} d_i \cdot B^{i-1}$

$\vec{\mathsf{r}}^{(\mu)}, \vec{\mathsf{r}}^{(\nu)} \xleftarrow{\$} \mathbb{Z}_p^{\tilde{J}}, \vec{\mathsf{r}}^{(\Omega)}, \vec{\mathsf{r}}^{(T)}, \vec{\mathsf{r}}^{(F)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}, \vec{\mathsf{r}}^{(M)} \xleftarrow{\$} \mathbb{Z}_p^{*B}, \mathsf{r}_{\tilde{K}}^{(\Omega)} \triangleq \mathsf{r}_\omega - \sum_{k\in[\tilde{K}-1]} \mathsf{r}_k^{(\Omega)}$

$R, S, U \xleftarrow{\$} \mathbb{Z}_p^*, \quad \left(\tilde{w}_{j,k} \triangleq \hat{d}_{j,k} \cdot \hat{B}_{j,k}\right)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}, \quad \left(m_c \triangleq \sum_{i\in[\tilde{N}]} \mathbb{1}(d_i = c)\right)_{c=0}^{B-1}$

$\mathcal{P}\Rightarrow\mathcal{V}: \left(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \tilde{w}_{j,k}} \cdot Q^{\mathsf{r}_k^{(\Omega)}}\right)_{k\in[\tilde{K}]}, \quad \left(M_c \triangleq G^{m_c} \cdot Q^{\mathsf{r}_c^{(M)}}\right)_{c=0}^{B-1}$

$\tilde{R} \triangleq G^{\sum_{j\in[\tilde{J}]} \mathsf{r}_j^{(\nu)}} \cdot Q^{\mathsf{r}_R} \in \mathbb{G}, \quad \tilde{S} \triangleq \prod_{j\in[\tilde{J}]} H_j^{\mathsf{r}_j^{(\mu)} \cdot \mathsf{r}_j^{(\nu)}} \cdot Q^{\mathsf{r}_S} \in \mathbb{G},$

$\tilde{U} \triangleq G^{\sum_{j\in[\tilde{J}]} B^{j-1} \cdot \mathsf{r}_j^{(\mu)}} \cdot Q^{\mathsf{r}_U}$

$\mathcal{P}\Leftarrow\mathcal{V}: \alpha \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P}: \left(f_{j,k} \triangleq \frac{1}{\alpha + \hat{d}_{j,k}}\right)_{(j,k)\in\mathcal{B}},$

$\tau_{j,k} \triangleq \begin{cases} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot \mathsf{r}_j^{(\nu)} + (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \cdot \mathsf{r}_j^{(\mu)}, & \text{if } (j,k) \in \mathcal{B} \\ 0, & \text{if } (j,k) \notin \mathcal{B} \end{cases}$

$\hat{\tau}_{k,k'}^{(j)} \triangleq \begin{cases} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot (\alpha \cdot \hat{B}_{j,k'} + \tilde{w}_{j,k'}) + \\ \hat{B}_{j,k'}^{-1} \cdot f_{j,k'} \cdot (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}), & \text{if } (j,k) \in \mathcal{B} \wedge (j,k') \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases}$

$\mathcal{P}\Rightarrow\mathcal{V}: \left(F_k \triangleq G^{\sum_{j\in[\tilde{J}]:(j,k)\in\mathcal{B}} f_{j,k}} \cdot Q^{\mathsf{r}_k^{(F)}}\right)_{k\in[\tilde{K}]}, \quad \left(\tilde{T}_k \triangleq \prod_{j\in[\tilde{J}]} H_j^{\tau_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\right)_{k\in[\tilde{K}]}$

$\left(\hat{T}_{k,k'} \triangleq \prod_{j\in[J]} H_j^{\hat{\tau}_{k,k'}^{(j)}} \cdot Q^{\mathsf{r}_{k,k'}^{(\hat{T})}}\right)_{k\in[\tilde{K}],k'\in[\tilde{K}]\setminus\{k\}}$

$\mathcal{P}\Leftarrow\mathcal{V}: \vec{\epsilon} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}$

$\mathcal{P}\Rightarrow\mathcal{V}: \left(v_j \triangleq \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} (\alpha \cdot \hat{B}_{j,k} + \tilde{w}_{j,k}) \cdot \epsilon_k + \mathsf{r}_j^{(\nu)} \in \mathbb{Z}_p\right)_{j\in[\tilde{J}]}$

$\left(\mu_j \triangleq \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \hat{B}_{j,k}^{-1} \cdot f_{j,k} \cdot \epsilon_k + \mathsf{r}_j^{(\mu)}\right)_{j\in[\tilde{J}]}$

$\tilde{\eta}_1 \triangleq \sum_{k\in[K],k'\in[K]\setminus\{k\}} \mathsf{r}_{k,k'}^{(\hat{T})} \epsilon_k \epsilon_{k'} + \vec{\mathsf{r}}^{(T)} \cdot \vec{\epsilon} + \mathsf{r}_S \in \mathbb{Z}_p, \quad \tilde{\eta}_4 \triangleq \vec{\mathsf{r}}^{(\Omega)} \cdot \vec{\epsilon} + \mathsf{r}_R$

$\tilde{\eta}_2 \triangleq \mathsf{r}_1^{(F)} \epsilon_1 + \sum_{k=2}^{\tilde{K}} \mathsf{r}_k^{(F)} \cdot \epsilon_k \cdot B^{-\tilde{J}(k-2)} - \tilde{\eta} + \mathsf{r}_U, \ \tilde{\eta}_3 \triangleq \sum_{c=0}^{B-1} \frac{\mathsf{r}_c^{(M)}}{\alpha + c} - \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(F)}$

$\mathcal{V}: \textsc{Check}$

$\prod_{j\in[\tilde{J}]} H_j^{(v_j \cdot \mu_j + \sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}} \epsilon_k^2)} \cdot Q^{\tilde{\eta}_1}$

$\stackrel{?}{=} H_{\tilde{J}}^{\sum_{k\in[\tilde{K}]} \epsilon_k^2} \cdot \prod_{k\in[\tilde{K}],k'\in[\tilde{K}]\setminus\{k\}} \hat{T}_{k,k'}^{\epsilon_k \epsilon_{k'}} \cdot \prod_{k\in[\tilde{K}]} \tilde{T}_k^{\epsilon_k} \cdot \tilde{S}$

$G^{\sum_{j\in[\tilde{J}]} B^{j-1} \cdot \mu_j} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} (F_1)^{\epsilon_1} \cdot \prod_{k=2}^{\tilde{K}} (F_k)^{\epsilon_k \cdot B^{-\tilde{J}(k-2)}} \cdot \tilde{U}$

$\prod_{k\in[\tilde{K}]} F_k \cdot Q^{\tilde{\eta}_3} \stackrel{?}{=} \prod_{c=0}^{B-1} M_c^{\frac{1}{\alpha+c}}$

$G^{\sum_{j\in[\tilde{J}]} v_j - \sum_{k\in[\tilde{K}]} \alpha \cdot \hat{B}_k \cdot \epsilon_k} \cdot Q^{\tilde{\eta}_4} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (\Omega_k)^{\epsilon_k} \cdot \tilde{R}$

$\mathtt{Cm}(\omega) \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k$

G.2 Aggregating Type-2 Range Arguments

By extending the type-2 range argument protocol, we can construct the aggregated type-2 range argument protocol in Fig. 11.

Figure 11: Aggregated VeRange type-2 range argument protocol

$\Pi_{\text{a.ty2}}\Big[\big(\mathtt{Cm}(\omega^{(t)}) \in \mathbb{G}\big)_{t\in[T]};\ (\omega^{(t)} \in \mathbb{Z}_p, \mathsf{r}_{\omega(t)} \in \mathbb{Z}_p^*)_{t\in[T]}\Big]$

SETUP : $H_{\tilde{J}} \triangleq \prod_{j\in[\tilde{J}]} H_j$

$\mathcal{P} \Leftarrow \mathcal{V} : \gamma \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} : \vec{\mathbf{d}}^{(t)} \in (\{0,...,B-1\})^{\tilde{N}}$ is the B-ary digit decompo. of $\omega : \omega^{(t)} = \sum_{i\in[\tilde{N}]} d_i^{(t)} \cdot B^{i-1}$

$\Big(\mathsf{r}_{j,k}^{(\mu)}, \mathsf{r}_{j,k}^{(\nu)} \xleftarrow{\$} \mathbb{Z}_p\Big)_{j\in[\tilde{J}],k\in[\tilde{K}]}$, $\vec{\mathsf{r}}^{(\Omega)}, \vec{\mathsf{r}}^{(T)}, \vec{\mathsf{r}}^{(V)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}$, $\vec{\mathsf{r}}^{(M)} \xleftarrow{\$} \mathbb{Z}_p^{*B}$

$\mathsf{r}_R, \mathsf{r}_S \xleftarrow{\$} \mathbb{Z}_p^*$, $\mathsf{r}_{\tilde{K}}^{(\Omega)} \triangleq \sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} - \sum_{k\in[\tilde{K}-1]} \mathsf{r}_k^{(\Omega)}$

$\Big(\bar{\tilde{w}}_{j,k} \triangleq \bar{\tilde{d}}_{j,k} \cdot \bar{\tilde{B}}_{j,k}\Big)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$, $\Big(m_c \triangleq \sum_{i\in[\tilde{N}]} \mathbb{1}(d_i = c)\Big)_{c=0}^{B-1}$

$\mathcal{P} \Rightarrow \mathcal{V} : \Big(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \bar{\tilde{w}}_{j,k}} \cdot Q^{\mathsf{r}_k^{(\Omega)}}\Big)_{k\in[\tilde{K}]}, \Big(M_c \triangleq G^{m_c} \cdot Q^{\mathsf{r}_c^{(M)}}\Big)_{c=0}^{B-1}$

$\Big(V_k \triangleq G^{\sum_{(j,k)\in\mathcal{B}} \bar{\tilde{B}}_{j,k} \cdot \mathsf{r}_{j,k}^{(\mu)}} \cdot Q^{\mathsf{r}_k^{(V)}}\Big)_{k\in[\tilde{K}]}$

$\tilde{R} \triangleq G^{\sum_{(j,k)\in\mathcal{B}} \mathsf{r}_{j,k}^{(\nu)}} \cdot Q^{\mathsf{r}_R}$, $\tilde{S} \triangleq \prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \mathsf{r}_{j,k}^{(\mu)} \cdot \mathsf{r}_{j,k}^{(\nu)}} \cdot Q^{\mathsf{r}_S}$

$\mathcal{P} \Leftarrow \mathcal{V} : \alpha \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} : \Big(f_{j,k} \triangleq \frac{1}{\alpha + \bar{\tilde{d}}_{j,k}}\Big)_{(j,k)\in\mathcal{B}}$

$\tau_{j,k} \triangleq \begin{cases} \bar{\tilde{B}}_{j,k}^{-1} \cdot f_{j,k} \cdot \mathsf{r}_{j,k}^{(\nu)} + (\alpha \cdot \bar{\tilde{B}}_{j,k} + \bar{\tilde{w}}_{j,k}) \cdot \mathsf{r}_{j,k}^{(\mu)}, & \text{if } (j,k) \in \mathcal{B} \\ 0, & \text{if } (j,k) \notin \mathcal{B} \end{cases}$

$\mathcal{P} \Rightarrow \mathcal{V} : \Big(\tilde{T}_k \triangleq \prod_{j\in[\tilde{J}]} H_j^{\tau_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\Big)_{k\in[\tilde{K}]}$

$\mathcal{P} \Leftarrow \mathcal{V} : \vec{e} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}$

$\mathcal{P} \Rightarrow \mathcal{V} : \Big(v_{j,k} \triangleq (\alpha \cdot \bar{\tilde{B}}_{j,k} + \bar{\tilde{w}}_{j,k}) \cdot \epsilon_k + \mathsf{r}_{j,k}^{(\nu)},\quad \mu_{j,k} \triangleq \bar{\tilde{B}}_{j,k}^{-1} \cdot f_{j,k} \cdot \epsilon_k + \mathsf{r}_{j,k}^{(\mu)}\Big)_{(j,k)\in\mathcal{B}}$

$\tilde{\eta}_1 \triangleq \vec{\mathsf{r}}^{(T)} \cdot \vec{e} + \mathsf{r}_S,\quad \tilde{\eta}_2 \triangleq \sum_{c=0}^{B-1} \frac{\mathsf{r}_c^{(M)}}{\alpha + c} + \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(V)} \cdot \epsilon_k^{-1},\quad \tilde{\eta}_3 \triangleq \vec{\mathsf{r}}^{(\Omega)} \cdot \vec{e} + \mathsf{r}_R$

$\mathcal{V} : \text{CHECK} \begin{cases} \prod_{j\in[\tilde{J}]} H_j^{\sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} v_{j,k}\cdot\mu_{j,k} + \sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}} \epsilon_k^2} \cdot Q^{\tilde{\eta}_1} \\ \qquad \stackrel{?}{=} H_{\tilde{J}}^{\sum_{k\in[\tilde{K}]} \epsilon_k^2} \cdot \prod_{k\in[\tilde{K}]} \tilde{T}_k^{\epsilon_k} \cdot \tilde{S} \\ G^{\sum_{(j,k)\in\mathcal{B}} \bar{\tilde{B}}_{j,k}\cdot\mu_{j,k}\cdot\epsilon_k^{-1}} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} \prod_{c=0}^{B-1} M_c^{\frac{1}{\alpha+c}} \cdot \prod_{k\in[\tilde{K}]} V_k^{\epsilon_k^{-1}} \\ G^{\sum_{(j,k)\in\mathcal{B}} v_{j,k} - \alpha\cdot\bar{\tilde{B}}_{j,k}\cdot\epsilon_k} \cdot Q^{\tilde{\eta}_3} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k^{\epsilon_k} \cdot \tilde{R} \\ \prod_{t\in[T]} \big(\mathtt{Cm}(\omega^{(t)})\big)^{\gamma^t} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \end{cases}$

Finally, we can also construct the aggregated type-3 range argument protocol in a similar manner as the aggregated type-2 range argument protocol.

Figure 12: Aggregated VeRange type-2B range argument protocol

$\Pi_{\text{a.ty2b}}\Big[\big(\mathtt{Cm}(\omega^{(t)}) \in \mathbb{G}\big)_{t\in[T]};\ (\omega^{(t)} \in \mathbb{Z}_p, \mathsf{r}_{\omega(t)} \in \mathbb{Z}_p^*)_{t\in[T]}\Big]$

SETUP : $\bar{\tilde{B}}_k \triangleq \sum_{\tilde{j}\in[\tilde{J}]:(j,k)\in\mathcal{B}} \bar{\tilde{B}}_{j,k}$, $H_{\tilde{j}} \triangleq \prod_{j\in[\tilde{J}]} H_j$

$\mathcal{P} \Leftarrow \mathcal{V} : \gamma \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} : \vec{\mathbf{d}}^{(t)} \in (\{0,...,B-1\})^{\tilde{N}}$ is the B-ary digit decompo. of $\omega : \omega^{(t)} = \sum_{i\in[\tilde{N}]} d_i^{(t)} \cdot B^{i-1}$

$\vec{\mathsf{r}}^{(\mu)}, \vec{\mathsf{r}}^{(\nu)} \xleftarrow{\$} \mathbb{Z}_p^{\tilde{J}}$, $\vec{\mathsf{r}}^{(\Omega)}, \vec{\mathsf{r}}^{(T)}, \vec{\mathsf{r}}^{(F)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}$, $\vec{\mathsf{r}}^{(M)} \xleftarrow{\$} \mathbb{Z}_p^{*B}$

$\mathsf{r}_{\tilde{K}}^{(\Omega)} \triangleq \sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} - \sum_{k\in[\tilde{K}-1]} \mathsf{r}_k^{(\Omega)}$

$R, S, U \xleftarrow{\$} \mathbb{Z}_p^*$, $\Big(\widetilde{w}_{j,k} \triangleq \bar{\tilde{d}}_{j,k} \cdot \bar{\tilde{B}}_{j,k}\Big)_{j=1,k=1}^{\tilde{J}\ \tilde{K}}$, $\Big(m_c \triangleq \sum_{i\in[\tilde{N}]} \mathbb{1}(d_i = c)\Big)_{c=0}^{B-1}$

$\mathcal{P} \Rightarrow \mathcal{V} : \Big(\Omega_k \triangleq G^{\sum_{j\in[\tilde{J}]} \widetilde{w}_{j,k}} \cdot Q^{\mathsf{r}_k^{(\Omega)}}\Big)_{k\in[\tilde{K}]}$, $\Big(M_c \triangleq G^{m_c} \cdot Q^{\mathsf{r}_c^{(M)}}\Big)_{c=0}^{B-1}$

$\tilde{R} \triangleq G^{\sum_{j\in[\tilde{J}]} \mathsf{r}_j^{(\nu)}} \cdot Q^{\mathsf{r}_R} \in \mathbb{G}$, $\tilde{S} \triangleq \prod_{j\in[\tilde{J}]} H_j^{\mathsf{r}_j^{(\mu)} \cdot \mathsf{r}_j^{(\nu)}} \cdot Q^{\mathsf{r}_S} \in \mathbb{G}$,

$\tilde{U} \triangleq G^{\sum_{j\in[\tilde{J}]} B^{j-1} \cdot \mathsf{r}_j^{(\mu)}} \cdot Q^{\mathsf{r}_U}$

$\mathcal{P} \Leftarrow \mathcal{V} : \alpha \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} : \Big(f_{j,k} \triangleq \frac{1}{\alpha + \bar{\tilde{d}}_{j,k}}\Big)_{(j,k)\in\mathcal{B}}$,

$\tau_{j,k} \triangleq \begin{cases} \bar{\tilde{B}}_{j,k}^{-1} \cdot f_{j,k} \cdot \mathsf{r}_j^{(\nu)} + (\alpha \cdot \bar{\tilde{B}}_{j,k} + \widetilde{w}_{j,k}) \cdot \mathsf{r}_j^{(\mu)}, & \text{if } (j,k) \in \mathcal{B} \\ 0, & \text{if } (j,k) \notin \mathcal{B} \end{cases}$

$\hat{\tau}_{k,k'}^{(j)} \triangleq \begin{cases} \bar{\tilde{B}}_{j,k}^{-1} \cdot f_{j,k} \cdot (\alpha \cdot \bar{\tilde{B}}_{j,k'} + \widetilde{w}_{j,k'}) + \\ \bar{\tilde{B}}_{j,k'}^{-1} \cdot f_{j,k'} \cdot (\alpha \cdot \bar{\tilde{B}}_{j,k} + \widetilde{w}_{j,k}), & \text{if } (j,k) \in \mathcal{B} \wedge (j,k') \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases}$

$\mathcal{P} \Rightarrow \mathcal{V} : \Big(F_k \triangleq G^{\sum_{j\in[\tilde{J}]:(j,k)\in\mathcal{B}} f_{j,k}} \cdot Q^{\mathsf{r}_k^{(F)}}\Big)_{k\in[\tilde{K}]}$, $\Big(\tilde{T}_k \triangleq \prod_{j\in[\tilde{J}]} H_j^{\tau_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\Big)_{k\in[\tilde{K}]}$

$\Big(\hat{T}_{k,k'} \triangleq \prod_{j\in[J]} H_j^{\hat{\tau}_{k,k'}^{(j)}} \cdot Q^{\mathsf{r}_{k,k'}^{(\hat{T})}}\Big)_{k\in[\tilde{K}],k'\in[\tilde{K}]\setminus\{k\}}$

$\mathcal{P} \Leftarrow \mathcal{V} : \vec{e} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}}$

$\mathcal{P} \Rightarrow \mathcal{V} : \Big(v_j \triangleq \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} (\alpha \cdot \bar{\tilde{B}}_{j,k} + \widetilde{w}_{j,k}) \cdot \epsilon_k + \mathsf{r}_j^{(\nu)} \in \mathbb{Z}_p\Big)_{j\in[\tilde{J}]}$

$\Big(\mu_j \triangleq \sum_{k\in[\tilde{K}]:(j,k)\in\mathcal{B}} \bar{\tilde{B}}_{j,k}^{-1} \cdot f_{j,k} \cdot \epsilon_k + \mathsf{r}_j^{(\mu)}\Big)_{j\in[\tilde{J}]}$

$\tilde{\eta}_1 \triangleq \sum_{k\in[K],k'\in[K]\setminus\{k\}} \mathsf{r}_{k,k'}^{(\hat{T})} \epsilon_k \epsilon_{k'} + \vec{\mathsf{r}}^{(T)} \cdot \vec{e} + \mathsf{r}_S \in \mathbb{Z}_p,\quad \tilde{\eta}_4 \triangleq \vec{\mathsf{r}}^{(\Omega)} \cdot \vec{e} + \mathsf{r}_R$

$\tilde{\eta}_2 \triangleq \mathsf{r}_1^{(F)} \epsilon_1 + \sum_{k=2}^{\tilde{K}} \mathsf{r}_k^{(F)} \cdot \epsilon_k \cdot B^{-\tilde{J}(k-2)-\tilde{\eta}} + \mathsf{r}_U,\ \tilde{\eta}_3 \triangleq \sum_{c=0}^{B-1} \frac{\mathsf{r}_c^{(M)}}{\alpha + c} - \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(F)}$

$\mathcal{V} : \text{CHECK} \begin{cases} \prod_{j\in[\tilde{J}]} H_j^{(v_j\cdot\mu_j + \sum_{k\in[\tilde{K}]:(j,k)\notin\mathcal{B}} \epsilon_k^2)} \cdot Q^{\tilde{\eta}_1} \\ \qquad \stackrel{?}{=} H_{\tilde{j}}^{\sum_{k\in[\tilde{K}]} \epsilon_k^2} \cdot \prod_{k\in[\tilde{K}],k'\in[\tilde{K}]\setminus\{k\}} \hat{T}_{k,k'}^{\epsilon_k\epsilon_{k'}} \cdot \prod_{k\in[\tilde{K}]} \tilde{T}_k^{\epsilon_k} \cdot \tilde{S} \\ G^{\sum_{j\in[\tilde{J}]} B^{j-1}\cdot\mu_j} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} (F_1)^{\epsilon_1} \cdot \prod_{k=2}^{\tilde{K}} (F_k)^{\epsilon_k \cdot B^{-\tilde{J}(k-2)-\tilde{\eta}}} \cdot \tilde{U} \\ \prod_{k\in[\tilde{K}]} F_k \cdot Q^{\tilde{\eta}_3} \stackrel{?}{=} \prod_{c=0}^{B-1} M_c^{\frac{1}{\alpha+c}} \\ G^{\sum_{j\in[\tilde{J}]} v_j - \sum_{k\in[\tilde{K}]} \alpha\cdot\bar{\tilde{B}}_k\cdot\epsilon_k} \cdot Q^{\tilde{\eta}_4} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (\Omega_k)^{\epsilon_k} \cdot \tilde{R} \\ \prod_{t\in[T]} \big(\mathtt{Cm}(\omega^{(t)})\big)^{\gamma^t} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \end{cases}$

Figure 13: Aggregated VeRange type-3 range argument protocol

$\Pi_{\text{a.ty3}}\Big[\big(\mathtt{Cm}(\omega^{(t)}) \in \mathbb{G}\big)_{t\in[T]};\ (\omega^{(t)} \in \mathbb{Z}_p, \mathsf{r}_{\omega(t)} \in \mathbb{Z}_p^*)_{t\in[T]}\Big]$

Setup : Distinct $z_0, z_1, ..., z_{\tilde{K}} \in \mathbb{Z}_p$

$$\mathsf{L}_k[X] \triangleq \prod_{k'\in\{0,...,\tilde{K}\}\setminus\{k\}} \frac{X - z_{k'}}{z_k - z_{k'}},\ \mathsf{L}_0[X] \triangleq \prod_{k\in[\tilde{K}]} (X - z_k),$$

$$\bar{B}_j[X] \triangleq \sum_{k\in[\tilde{K}]} \bar{\bar{B}}_{j,k} \cdot \mathsf{L}_k[X]$$

$\mathcal{P} \Leftarrow \mathcal{V} : \gamma \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} : \vec{\mathbf{d}}^{(t)} \in (\{0, ..., B-1\})^{\tilde{N}}$ is the B-ary digit decompo. of $\omega : \omega^{(t)} = \sum_{i\in[\tilde{N}]} d_i^{(t)} \cdot B^{i-1}$

$$\vec{\mathsf{r}}^{(\mathrm{d})} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{J}},\quad \vec{\mathsf{r}}^{(\Omega)}, \vec{\mathsf{r}}^{(D)} \xleftarrow{\$} \mathbb{Z}_p^{*\tilde{K}},\quad \mathsf{s}, \mathsf{r}_1, \mathsf{r}_2 \xleftarrow{\$} \mathbb{Z}_p^*$$

$$\mathsf{r}_{\tilde{K}}^{(\Omega)} \triangleq \sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} - \sum_{k\in[\tilde{K}-1]} \mathsf{r}_k^{(\Omega)}$$

$$\big(\widetilde{w}_{j,k} \triangleq \bar{\bar{d}}_{j,k} \cdot \bar{\bar{B}}_{j,k} \in \mathbb{Z}_p\big)_{j=1,k=1}^{\tilde{J}\quad\tilde{K}},\quad \bar{d}_j[X] \triangleq \mathsf{r}_j^{(\mathrm{d})} \cdot \mathsf{L}_0[X] + \sum_{k\in[\tilde{K}]} \bar{\bar{d}}_{j,k} \cdot \mathsf{L}_k[X]$$

$$\mathsf{B}_j[X] \triangleq \frac{\bar{d}_j[X] \cdot (\bar{d}_j[X]-1) \cdots (\bar{d}_j[X]-B+1)}{\mathsf{L}_0[X]}$$

$$\Big(\widetilde{w}_k \triangleq \sum_{j\in\tilde{J}} \widetilde{w}_{j,k}\Big)_{k\in\tilde{K}},\quad \mathsf{S}[X] \triangleq \mathsf{s} + \frac{\sum_{k\in[\tilde{K}]} \widetilde{w}_k \cdot \mathsf{L}_k[X] - \sum_{j\in[\tilde{J}]} \bar{d}_j[X] \cdot \bar{B}_j[X]}{\mathsf{L}_0[X]}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : \Big(D_k \triangleq \prod_{j\in[\tilde{J}]} G_j^{\bar{\bar{d}}_{j,k}} \cdot Q^{\mathsf{r}_k^{(D)}}\Big)_{k\in[\tilde{K}]},\ R_1 \triangleq \prod_{j\in[\tilde{J}]} G_j^{\mathsf{r}_j^{(\mathrm{d})}} \cdot Q^{\mathsf{r}_1},\ R_2 \triangleq G^{\mathsf{s}} \cdot Q^{\mathsf{r}_2}$$

$$\Big(\Omega_k \triangleq G^{\widetilde{w}_k} \cdot Q^{\mathsf{r}_k^{(\Omega)}}\Big)_{k\in[\tilde{K}]},\qquad \mathtt{Cm}_{\mathsf{S}} \triangleq \mathtt{PolyCm}_{\mathtt{BCCGP}}[\mathsf{S}] \in \mathbb{G}^{U+1}$$

$\mathcal{P} \Leftarrow \mathcal{V} : \beta \xleftarrow{\$} \mathbb{Z}_p^*$

$$\mathcal{P} : \mathsf{B}[X] \triangleq \sum_{j\in[\tilde{J}]} \beta^j \cdot \mathsf{B}_j[X]$$

$\mathcal{P} \Rightarrow \mathcal{V} : \mathtt{Cm}_{\mathsf{B}} \triangleq \mathtt{PolyCm}_{\mathtt{BCCGP}}[\mathsf{B}] \in \mathbb{G}^{U+1}$

$\mathcal{P} \Leftarrow \mathcal{V} : x \xleftarrow{\$} \mathbb{Z}_p \setminus \{z_0, z_1, ..., z_{\tilde{K}}\}$

$\mathcal{P} \Rightarrow \mathcal{V} : (\bar{d}_j \triangleq \bar{d}_j[x] \in \mathbb{Z}_p)_{j\in[\tilde{J}]},\quad y_{\mathsf{B}} \triangleq \mathsf{B}[x],\quad \pi_{\mathsf{B}} \triangleq \mathtt{PolyEv}_{\mathtt{BCCGP}}[\mathsf{B}, x] \in \mathbb{Z}_p^{V+1}$

$$y_{\mathsf{S}} \triangleq \mathsf{S}[x] \in \mathbb{Z}_p,\quad \pi_{\mathsf{S}} \triangleq \mathtt{PolyEv}_{\mathtt{BCCGP}}[\mathsf{S}, x] \in \mathbb{Z}_p^{V+1}$$

$$\tilde{\eta}_1 \triangleq \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(D)} \cdot \mathsf{L}_k[x] + \mathsf{r}_1 \cdot \mathsf{L}_0[x],\ \tilde{\eta}_2 \triangleq \sum_{k\in[\tilde{K}]} \mathsf{r}_k^{(\Omega)} \cdot \mathsf{L}_k[x] + \mathsf{r}_2 \cdot \mathsf{L}_0[x]$$

$$\mathcal{V} : (\bar{B}_j \triangleq \bar{B}_j[x] \in \mathbb{Z}_p)_{j\in[\tilde{J}]}$$

$$\textsc{Check}\ \begin{cases} \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_{\mathsf{B}}, x, y_{\mathsf{B}}, \pi_{\mathsf{B}}] \stackrel{?}{=} 1 \\ \mathtt{PolyVf}_{\mathtt{BCCGP}}[\mathtt{Cm}_{\mathsf{S}}, x, y_{\mathsf{S}}, \pi_{\mathsf{S}}] \stackrel{?}{=} 1 \\ \prod_{j\in[\tilde{J}]} G_j^{\bar{d}_j} \cdot Q^{\tilde{\eta}_1} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (D_k)^{\mathsf{L}_k[x]} \cdot R_1^{\mathsf{L}_0[x]} \\ \sum_{j\in[\tilde{J}]} \beta^j \cdot \bar{d}_j \cdots (\bar{d}_j - B + 1) \stackrel{?}{=} y_{\mathsf{B}} \cdot \mathsf{L}_0[x] \\ G^{(y_{\mathsf{S}} \cdot \mathsf{L}_0[x] + \sum_{j\in[\tilde{J}]} \bar{d}_j \cdot \bar{B}_j)} \cdot Q^{\tilde{\eta}_2} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} (\Omega_k)^{\mathsf{L}_k[x]} \cdot R_2^{\mathsf{L}_0[x]} \\ \prod_{t\in[T]} \big(\mathtt{Cm}(\omega^{(t)})\big)^{\gamma^t} \stackrel{?}{=} \prod_{k\in[\tilde{K}]} \Omega_k \end{cases}$$

Figure 14: Aggregated Flashproofs range argument protocol

$\Pi_{\text{a.flash}}\Big[\big(\mathtt{Cm}(\omega^{(t)}) \in \mathbb{G}\big)_{t\in[T]};\ (\omega^{(t)} \in \mathbb{Z}_p, \mathsf{r}_{\omega(t)} \in \mathbb{Z}_p^*)_{t\in[T]}\Big]$

$\mathcal{P} \Leftarrow \mathcal{V} : \gamma \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} : \vec{\mathbf{b}}^{(t)} \in \{0,1\}^N$ is the bit-decomposition of ω such that $\omega^{(t)} = \sum_{i\in[N]} b_i^{(t)} \cdot 2^{i-1}$

$$\vec{\mathsf{r}} \xleftarrow{\$} \mathbb{Z}_p^{*J},\ \vec{\mathsf{r}}^{(W)}, \vec{\mathsf{r}}^{(T)} \xleftarrow{\$} \mathbb{Z}_p^{*K},\ \vec{\mathsf{r}}^{(\hat{T})} \xleftarrow{\$} \mathbb{Z}_p^{*K(K\text{-}1)/2},\ \mathsf{r}_R, \mathsf{r}_S \xleftarrow{\$} \mathbb{Z}_p^*$$

$$\mathsf{r}_K^{(W)} \triangleq \sum_{t\in[T]} \gamma^t \cdot \mathsf{r}_{\omega(t)} - \sum_{k\in[K-1]} \mathsf{r}_k^{(W)}$$

$$\Big(\bar{\bar{w}}_{j,k} \triangleq \bar{\bar{b}}_{j,k} \cdot \bar{\bar{2}}_{j,k}\Big)_{j\in[J],k\in[K]},\ \Big(t_{j,k} \triangleq \mathsf{r}_j \cdot (\bar{\bar{2}}_{j,k} - 2\bar{\bar{w}}_{j,k})\Big)_{j\in[J],k\in[K]}$$

$$\Big(\hat{t}_{k,k'}^{(j)} \triangleq \sum_{j\in[J]} \bar{\bar{w}}_{j,k'} (\bar{\bar{2}}_{j,k} - \bar{\bar{w}}_{j,k}) + \bar{\bar{w}}_{j,k} (\bar{\bar{2}}_{j,k'} - \bar{\bar{w}}_{j,k'})\Big)_{k\in[K],k'\in[K]\setminus\{k\}}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : \Big(W_k \triangleq G^{\sum_{j\in[J]} \bar{\bar{w}}_{j,k}} \cdot Q^{\mathsf{r}_k^{(W)}}\Big)_{k\in[K]},\ \Big(T_k \triangleq \prod_{j\in[J]} H_j^{t_{j,k}} \cdot Q^{\mathsf{r}_k^{(T)}}\Big)_{k\in[K]}$$

$$\Big(\hat{T}_{k,k'} \triangleq \prod_{j\in[J]} H_j^{\hat{t}_{k,k'}^{(j)}} \cdot Q^{\mathsf{r}_{k,k'}^{(\hat{T})}}\Big)_{k\in[K],k'\in[K]\setminus\{k\}}$$

$$R \triangleq G^{\sum_{j\in[J]} \mathsf{r}_j} \cdot Q^{\mathsf{r}R} \in \mathbb{G},\qquad S \triangleq \prod_{j\in[J]} H_j^{-(\mathsf{r}_j)^2} \cdot Q^{\mathsf{r}S}$$

$\mathcal{P} \Leftarrow \mathcal{V} : \vec{\epsilon} \xleftarrow{\$} \mathbb{Z}_p^{*K}$

$$\mathcal{P} \Rightarrow \mathcal{V} : \Big(v_j \triangleq \sum_{k\in[K]} \bar{\bar{w}}_{j,k} \cdot \epsilon_k + \mathsf{r}_j\Big)_{j\in[J]},\quad \eta_2 \triangleq \vec{\mathsf{r}}^{(W)} \cdot \vec{\epsilon} + \mathsf{r}_R$$

$$\eta_1 \triangleq \sum_{k\in[K],k'\in[K]\setminus\{k\}} \mathsf{r}_{k,k'}^{(\hat{T})} \epsilon_k \epsilon_{k'} + \vec{\mathsf{r}}^{(T)} \cdot \vec{\epsilon} + \mathsf{r}_S$$

$$\mathcal{V} : \Big(u_j \triangleq \sum_{k\in[K]} \bar{\bar{2}}_{j,k} \cdot \epsilon_k - v_j\Big)_{j\in[J]}$$

$$\textsc{Check}\ \begin{cases} \prod_{j\in[J]} H_j^{v_j \cdot u_j} \cdot Q^{\eta_1} \stackrel{?}{=} \prod_{k\in[K],k'\in[K]\setminus\{k\}} \hat{T}_{k,k'}^{\epsilon_k \epsilon_{k'}} \cdot \prod_{k\in[K]} T_k^{\epsilon_k} \cdot S \\ G^{\sum_{j\in[J]} v_j} \cdot Q^{\eta_2} \stackrel{?}{=} \prod_{k\in[K]} W_k^{\epsilon_k} \cdot R \\ \prod_{t\in[T]} \big(\mathtt{Cm}(\omega^{(t)})\big)^{\gamma^t} \stackrel{?}{=} \prod_{k\in[K]} W_k \end{cases}$$