

Verifiable Weighted Secret Sharing based on Chinese Remainder Theorem

Xinyu Zhang¹, Weiping Ji¹, Tsz Hon Yuen¹, Ron Steinfeld¹, Joseph K. Liu¹,
and Shujie Cui¹

Monash University, Melbourne, Australia
{Xinyu.Zhang2, Weiping.Ji, John.TszHonYuen,
Ron.Steinfeld, Joseph.Liu, Shujie.Cui}@monash.edu

Abstract. In this work, we present the first practical verifiable weighted secret sharing (VWSS) scheme based on the Chinese Remainder Theorem (CRT). Classical secret sharing schemes, such as Shamir’s, assume participants with equal weights, which is inadequate for emerging applications like stake-based DAO voting and proof-of-stake blockchains, where parties naturally have unequal voting power or stakes. While prior work (e.g., Garg et al., Crypto’23) demonstrated that CRT-based (ramp) secret sharing can support weighted access structures more efficiently than linear secret sharing schemes, existing constructions only guarantee security against honest-but-curious dealers. In contrast to traditional linear secret sharing schemes, extending verifiability to CRT-based weighted secret sharing remains challenging, as current approaches either fail to support weighted access structures or incur prohibitive computational overhead on the dealer, limiting their practicality.

To address this gap, we develop novel Σ -protocols for the *unbounded* proof-of-mod (UPoM) relation using integer commitments in groups of unknown order, which may be of independent interest. Combining our UPoM protocols and new observations on the structural properties of CRT-based secret sharing, we construct a VWSS scheme in which the dealer broadcasts only $O(|A|)$ commitments to secret shares, where A is a subset of parties whose total weight meets the reconstruction threshold T . This improves upon the closest prior CRT-based VWSS scheme (Shehata et al., CVC’25), which requires the dealer to broadcast commitments to the secret shares of all parties. Concretely, for a system with four participants and total weight 524, our scheme generates VWSS proofs in around 0.2 seconds, *orders of magnitude faster* than the construction of CVC’25, highlighting its practicality for real-world applications.

Keywords: Chinese Remainder Theorem, Verifiable Weighted Secret Sharing, Groups of Unknown Order, Distributed Key Generation

1 Introduction

A secret sharing (SS) scheme allows a dealer to randomly divide a secret s into n shares, one for each party, such that any authorised subset of parties can reconstruct the secret, while any unauthorised subset learns nothing about it. Secret

sharing is a fundamental tool in many distributed cryptographic protocols, including secure multiparty computation (MPC) [26,20], threshold cryptography (e.g., threshold encryption [8,14] and signatures [9,15]), and distributed randomness generation [22]. Most existing secret sharing schemes, such as Shamir’s secret sharing [23], assume equally weighted shareholders. However, this assumption limits their applicability in many modern scenarios like stake-based DAO voting, Proof-of-Stake (PoS) blockchains [16] and Oracle networks [2], where parties naturally have unequal weights. In fact, Shamir’s original secret sharing paper briefly discussed extending the scheme to handle weighted participants through the technique commonly referred to as *virtualisation*, where a user with weight w acts as w virtual parties of unit weight. As a result, the user receives w secret shares, leading to communication and storage complexity of $O(w \cdot \ell)$, where $\ell = \lceil \log_2 |\mathbb{F}| \rceil$ is the bit-length of the share. When the field is chosen such that $\ell = \Theta(\lambda)$ for the computational security parameter λ , the complexity becomes $O(w \cdot \lambda)$. Notably, this approach becomes extremely inefficient in systems with a large weight gap between users.

To overcome the above limitation, Garg et al. [13] proposed an efficient weighted (ramp) secret sharing scheme¹ based on the Chinese Remainder Theorem (CRT), which achieves communication and storage complexity of $O(w)$. However, the scheme assumes an honest dealer during share distribution. Verifiable secret sharing (VSS) [5] strengthens secret sharing by allowing parties to verify the consistency between the received share and dealer’s secret. Existing VSS constructions, such as Feldman’s [10] and Pedersen’s [21], are designed for linear secret sharing schemes and do not naturally extend to CRT-based weighted secret sharing. As also noted in [13], constructing an *efficient* verifiable weighted secret sharing scheme remains an open problem.

1.1 Contributions

We present the first practical construction of *verifiable weighted secret sharing* based on the Chinese Remainder Theorem (CRT), suitable for real-world deployment. At the core of our approach are Σ -protocols for *unbounded proof-of-mod* relation, built on integer commitments in groups of unknown order with *transparent setup*. Our main contributions are summarised as follows:

- **Unbounded Proof-of-Mod (UPoM):** We present a Σ -protocol that allows the prover to convince the verifier that $s \equiv v \pmod{N}$ for committed s and v , where s is an arbitrary integer. In particular, given commitments to s and v , our protocol produces a compact proof consisting of two integer responses whose sizes depend on the bit length of s and v , two additional integer responses whose sizes depend on an upper bound on the group order, and an auxiliary relation enforcement in $\mathbb{Z}_N$. However, the soundness error depends on the size of the prime factors of N , and can be non-negligible when these factors are small.

¹ A ramp secret sharing scheme allows a gap between privacy and reconstruction threshold.

- **Unbounded Proof-of-Mod with Soundness Boost:** To avoid relying on parallel repetition for soundness amplification, which may incur impractical overhead, we further refine the aforementioned UPoM protocol so that its soundness error depends solely on the size of the challenge space. Our revised construction only has a slight increase in communication and computational complexity, adding one integer commitment in the group of unknown order, two additional integer responses, and the auxiliary relation enforcement over the integers (as opposed to $\mathbb{Z}_N$ in the original UPoM). The resulting protocol applies to arbitrary moduli N .
- **Verifiable Weighted Secret Sharing:** Building on both UPoM protocols, we construct a verifiable weighted secret sharing (VWSS) scheme. Rather than generating separate UPoM proofs for each user, we develop two novel insights into the structural properties of CRT-based secret sharing that enable the dealer to produce a *single* UPoM proof for all participants plus $O(|A|)$ commitments. Here, A is a subset of parties whose total weight meets the reconstruction threshold. We further show two approaches for setting up the public parameters of our CRT-based VWSS: one optimised for faster proof generation and smaller private messages, and the other optimised for faster verification and compact broadcast messages.

As shown in Table 1, using the weight distribution of the 63 leading Ethereum stakeholders reported in [24], the first approach achieves a broadcast communication cost of 22.92KB and total private communication of 7.71KB, with a (core) proof generation time of 1.034 seconds and proof verifies in 2.586 seconds. In contrast, the second approach produces 6.22KB broadcast messages and 81.57KB private messages in total, with a (core) proof generation time of 6.443 seconds and the proof verifies in 0.781 seconds.² Although the construction in [24] is more communication-efficient, its proof generation overhead is orders of magnitude slower than our schemes, making it impractical for real-world applications. Compared with Feldman’s VSS [10] with virtualisation, which incurs approximately 75KB of broadcast communication and 147KB of total private communication, assuming compressed elliptic curve group elements of 32 bytes (e.g., secp256k1), both of our constructions achieve substantially lower communication overheads.

To the best of our knowledge, our VWSS construction is the first practical solution to the open problem posed in [13] of designing a practical verifiable weighted secret sharing scheme for weighted cryptosystems.

- **Distributed Key Generation:** As a concrete application of our VWSS scheme, we construct a distributed key generation protocol for (EC)DSA keys. While our scheme works in groups of unknown order, (EC)DSA keys are generated in discrete logarithm groups of known prime order, leading to a natural group mismatch problem. We resolve the issue by integrating

² We report the core proving time for generating the UPoM proofs and the range proof. The three square decomposition required by the range proof of Couteau et al. [6] is measured separately because of its high runtime variability, where the average execution time is 3.056 seconds across 100 runs.

Scheme	Core Proving Time (s)	Verify Time (s)	Broadcast (KB)	Total Private (KB)
VWSS-Method 1	1.034	2.586	22.92	7.71
VWSS-Method 2	6.443	0.781	6.22	81.57

Table 1. Performance of the proposed VWSS instantiated with 63 leading Ethereum stakeholders, with a total weight 4711 and reconstruction threshold 2402.

an exponent equality check into the underlying UPoM protocol, incurring a negligible overhead which consists of a single additional discrete logarithm group element.

1.2 Related Works

Unlike linear secret sharing schemes such as Shamir’s, adding verifiability to non-linear schemes (e.g., CRT-based secret sharing) remains largely underexplored. One prior work [17] constructed a CRT-based verifiable secret sharing scheme using groups of unknown order with a trusted setup. However, their construction differs fundamentally from ours and does not extend to the weighted setting, as discussed below.

At a high-level, the construction in [17] embeds CRT shares into discrete logarithm (DLOG) groups. Let N be an integer whose prime factorisation is hard, and let $P = \prod_{i=1}^n p_i$ denote the product of the primes used in the CRT-based secret sharing scheme. The dealer first computes a unique integer $g \in \mathbb{Z}_P$ such that

$$g \equiv g_i \pmod{p_i},$$

where each $g_i \in \mathbb{Z}_{p_i}^*$ for $i \in [n]$. The key observation is that

$$g^s \pmod{PN} \equiv g_i^{v_i} \pmod{p_i} \iff v_i = s \pmod{p_i}.$$

The dealer thus only broadcasts $g^s \pmod{PN}$ together with a range proof to ensure the lifted CRT secret s is well-formed, while each participant locally verifies the share using (g_i, v_i) .

The limitation of this approach is that its security relies on the hardness of the discrete logarithm problem in each $\mathbb{Z}_{p_i}^*$. In particular, each modulus must satisfy $p_i = 2m_i + 1$ with sufficiently large prime m_i . However, in CRT-based weighted ramp secret sharing, the bit length of m_i is proportional to $c \cdot w_i$, where w_i is the weight of party i and c is a common scalar. Since some weights can be very small, the corresponding moduli is too small to provide meaningful DLOG security. One could address this issue by choosing c to be $O(\lambda)$, ensuring that all moduli are cryptographically large. However, this destroys the efficiency advantage of weighted ramp secret sharing, causing the communication and storage complexity to increase from $O(w)$ to $O(w \cdot \lambda)$.

A recent work by Shehata et al. [24] proposed a CRT-based verifiable weighted secret sharing scheme, addressing the open problem posed in [13]. Their construction is based on a *bounded* proof-of-mod protocol, a non-interactive zero-knowledge argument (NIZK) proving that two committed values v and s satisfy $v \equiv s \pmod{p}$ under the additional assumption that $s < q$, where q denotes the order of the underlying commitment group. The protocol is instantiated using a disjunction of Bulletproofs-based range proofs [3].

To support arbitrarily large values of s , their construction decomposes s into its base- q representation, thereby reducing the proof-of-mod relation for large integers into multiple bounded proof-of-mod instances. Consequently, the dealer broadcasts $O(n)$ commitments together with a proof of size $O(\log mn)$, where n denotes the number of users and m denotes the number of digits in the base- q decomposition of s . Although the construction achieves significantly smaller proof sizes than virtualisation-based weighted VSS derived from Shamir’s secret sharing, it suffers from a prohibitive prover-efficiency bottleneck, which scales linearly with $m \cdot n$. Concretely, their experimental evaluation shows that even for small parameters (e.g., decomposition size $m = 4$ and $n = 4$ parties), proof generation already requires more than five minutes. As also acknowledged by the authors, the prover time is too high to be used in practice.

2 Technical Overview

2.1 CRT-based Weighted Secret Sharing

We begin by recalling the Chinese Remainder Theorem (CRT)-based weighted ramp secret sharing scheme proposed in [13]. In the CRT-based secret sharing, the dealer holds a secret $s_0 \in \mathbb{Z}_{p_0}$ for some prime p_0 . To share s_0 among n parties, the dealer lifts the secret by computing $s = s_0 + u \cdot p_0$ for $u \xleftarrow{\$} [0, L)$, where L is the parameter that depends on the threshold. Therefore, the upper bound on s is $p_0 + Lp_0 = (L + 1)p_0$. Then, the dealer chooses n distinct primes $p_1, \dots, p_n$ and computes shares $v_i = s \pmod{p_i}$ for all $i \in [n]$.

Consider an authorised set of parties $A \subseteq [n]$ and $P_A = \prod_{i \in A} p_i$, by the Chinese Remainder Theorem, the system of equations $\{v_i = s \pmod{p_i}\}_{i \in A}$ has a unique solution (i.e., s) modulo P_A if $(L + 1)p_0 \leq P_A - 1$. On the other hand, let $\bar{A}$ be an unauthorised set of parties. The adversary’s view is then $\{v_i = s \pmod{p_i}\}_{i \in \bar{A}}$ and by CRT, it has a unique solution $s' \pmod{P_{\bar{A}}}$ for $P_{\bar{A}} = \prod_{i \in \bar{A}} p_i$. To ensure privacy, it must be shown that $s' \pmod{P_{\bar{A}}}$ is statistically close to the uniform distribution, which is indeed the case when $P_{\bar{A}}/L$ is exponentially small. Therefore, one can construct a CRT-based secret sharing by picking L such that

$$\max_{\bar{A}} P_{\bar{A}} \ll L \leq \min_A P_A / 2^\lambda, \quad (1)$$

assuming p_0 is a prime of λ (security parameter) bits.

To extend CRT-based secret sharing in the weighted setting, Garg et. al. [13] exploit the gap between the privacy threshold (t) and reconstruction threshold

(T). Particularly, let w_i be the weight of party i and $c \in \mathbb{N}$ be a constant scalar across all parties. To embed weights into each share, a natural solution is to choose p_i with $c \cdot w_i$ bits. Notably, we can associate weights, the reconstruction and the privacy thresholds as follows:

- If $A \subseteq [n]$ is an authorised set, then $\sum_{i \in A} w_i \geq T$.
- If $\bar{A} \subset [n]$ is an unauthorised set, then $\sum_{i \in \bar{A}} w_i \leq t$.

Now, let us denote $P_{\max} = \max_{\bar{A}} P_{\bar{A}}$ and $P_{\min} = \min_A P_A$. Given the prior Inequality (1), we can write

$$P_{\max} \cdot 2^\lambda \leq L \leq P_{\min} \cdot 2^{-\lambda},$$

where $P_{\max} = 2^{c \cdot \sum_{i \in \bar{A}} w_i} = 2^{c \cdot t}$ and $P_{\min} = 2^{c \cdot \sum_{i \in A} w_i} = 2^{c \cdot T}$.

Assuming the standard secret sharing setting where $T = t + 1$, we have

$$2^{c \cdot t} \cdot 2^\lambda \leq L \leq 2^{c \cdot T} \cdot 2^{-\lambda} \implies 2^{c(T-t)} \geq 2^{2\lambda} \implies c = O(\lambda),$$

which has the same complexity as the virtualisation-based approach for weighted secret sharing, since each secret share is $O(c \cdot w_i) = O(\lambda \cdot w_i)$ bits. To improve the efficiency, the authors propose to use *ramp* secret sharing with a sufficiently large gap between the reconstruction threshold and the privacy threshold (i.e., $T - t \geq 2\lambda$), leading to $c = 1$, where each secret share now is $O(w_i)$ bits.

2.2 Obstacles of Designing VWSS based on CRT

A natural approach for designing verifiable CRT-based secret sharing is to let the dealer commit to the secret and shares, then show that the committed shares are congruent to the secret modulo the corresponding prime. Notably, if $s \equiv v \pmod{p}$, then there exists $k \in \mathbb{Z}$ such that $s = kp + v$, leading to a linear relation among s, k, v . By exploiting the homomorphic properties of commitment schemes, this suggests a seemingly straightforward approach to constructing a verifiable CRT-based (weighted) secret sharing scheme.

However, as also observed in [24], commitment schemes based on discrete logarithm groups (e.g., Pedersen commitment [21]) incur a wrap-around issue modulo the commitment group order q . In particular, let c_s, c_k, c_v be commitments to s, k, v , respectively. Showing that $c_s = c_v \cdot c_k^p$ only establishes the fact that

$$s \equiv (v + kp) \pmod{q},$$

which does not imply $s \equiv v \pmod{p}$. This motivates the approach proposed in [24] (see Section 1.2 for a detailed discussion), where the prover uses range proofs to show that both sides of the congruence are bounded by q , thereby preventing the wrap-around.

2.3 Our Solution

We first observe that the aforementioned wrap-around issue in proving $s \equiv v \pmod p$ stems from the fact that the group order is known to the (malicious) prover. If, instead, we use an integer commitment in group $\mathbb{G}$ of unknown order [7,6,12], then the relation $s \equiv (v + kp) \pmod{|\mathbb{G}|}$ necessarily implies $s = v + kp$. Otherwise, it would violate the order assumption in groups of unknown order (GUOs). Although GUOs typically feature with a trusted setup, recent works [6,12] show that integer commitment schemes in GUOs can be constructed with *transparent setup* as well. This comes at the cost of relaxing the statistical hiding to computational hiding under the decisional cross-group Diffie-Hellman assumption [6] or the subgroup indistinguishability assumption [12].

Given integer commitments to s and v , there is a standard Σ -protocol between a prover and a verifier for proving knowledge of an opening (a, r_a) to a commitment $c_a = \text{Commit}(a; r_a)$, where $a \in \{s, v\}$, as follows:

- Prover picks $(\bar{a}, \bar{r}_a)$ and sends $\bar{c}_a = \text{Commit}(\bar{a}, \bar{r}_a)$.
- Verifier sends a challenge $\gamma \xleftarrow{\$} [0, C)$.
- Prover sends the response $R_a = \bar{a} + \gamma \cdot a$ and $R_{r_a} = \bar{r}_a + \gamma \cdot r_a$.
- Verifier accepts if $\text{Commit}(R_a, R_{r_a}) = \bar{c}_a \cdot c_a^\gamma$.

Building on the knowledge of opening protocol, we observe that if $s \equiv v \pmod p$, then $R_s - R_v \equiv \bar{s} - \bar{v} \pmod p$, conditioned on the choice of γ .³ This allows the prover to enforce the congruence relation between s and v (modulo p) by computing $\Delta = \bar{s} - \bar{v} \pmod p$ and sending it to the verifier along with $(\bar{c}_s, \bar{c}_v)$, leading to an unbounded proof-of-mod (UPoM) protocol.

However, there are two key obstacles. First, since integer commitment schemes in [6,12] allow *relaxed rational openings*, the extractor in the knowledge soundness proof of UPoM can only recover rational witnesses for the proof-of-mod relation. Second, the soundness of UPoM depends on the size of the smallest prime factor of p as there may exist multiple choices of $\gamma \xleftarrow{\$} [0, C)$ such that $R_s - R_v \equiv \bar{s} - \bar{v} \pmod p$.

We address the first issue following [12] by working in the generic group model. That is, if the adversary is generic, the extracted rational witnesses are in fact integers. To resolve the second issue, we require the prover to additionally commit to $k \in \mathbb{Z}$ where $s = v + kp$, and prove knowledge of the opening of c_k . With this modification, the relation enforcement Δ becomes $\Delta = \bar{s} - \bar{v} - \bar{k}p$, which is checked over integers rather than modulo p . This removes the ambiguity arising from multiple valid choices of γ , making the soundness error depends solely on the challenge space. The overhead is only modest, including two additional commitments $(c_k, \bar{c}_k)$ and two additional responses (R_k, R_{r_k}) , without resorting to the naive parallel executions that can incur significantly higher costs.

Finally, we use UPoM protocols to construct a verifiable weighted secret sharing (VWSS) scheme. A straightforward approach would be for the dealer to

³ We note that p is not necessarily a prime, but we keep this notation for consistency with previous discussions.

commit to the lifted CRT secret s and invoke UPoM for each individual share v_i , proving that $s \equiv v_i \pmod{p_i}$ for all $i \in [n]$. While this yields a constant-size broadcast message, it requires the dealer to generate n UPoM proofs and send the individual proof message to each user via private channels.

We present two novel observations that can substantially reduce the overhead for the dealer. In particular, our approach requires the dealer to broadcast $O(|A|)$ commitments for an authorised subset of parties A , but significantly reduces the size of private messages and the proving time. Our first observation is that, in CRT-based secret sharing, there exists a public *global* set of CRT-coefficients $(\lambda_i)^4$, such that for any sufficiently weighted subsets A and $\hat{A}$,

$$s = \sum_{i \in A} \lambda_i v_i \pmod{\prod_{i \in A} p_i} = \sum_{i \in \hat{A}} \lambda_i v_i \pmod{\prod_{i \in \hat{A}} p_i},$$

and for any $i \in A \cap \hat{A}$, the coefficient λ_i remains unchanged.

Now, fix a sufficiently weighted subset $A_0 \subseteq [n]$ and let $A_0' = [n] \setminus A_0$. We observe that if $\sum_{i \in A_0} \lambda_i v_i \equiv s \pmod{\prod_{i \in A_0} p_i}$, then it follows that $s \equiv v_i \pmod{p_i}$ since $\lambda_i \equiv 0 \pmod{p_j}$ for all $j \neq i$ and $\lambda_i = 1 \pmod{p_i}$. Consequently, for users in A_0 , it suffices for the dealer to prove a single relation $s \equiv v \pmod{P_{A_0}}$, where $v = \sum_{i \in A_0} \lambda_i v_i$ and $P_{A_0} = \prod_{i \in A_0} p_i$, leading to a single UPoM proof for all users $i \in A_0$.

More interestingly, for any $j \in A_0'$, the global nature of CRT-coefficients used in previous proof implies that

$$s \equiv v + \lambda_j v_j \pmod{P_{A_0} \cdot p_j} \iff s - v \equiv \lambda_j v_j \pmod{p_j} \iff s \equiv v_j \pmod{p_j}.$$

The first equivalence holds since $\gcd(P_{A_0}, p_j) = 1$, and $s \equiv \lambda_j v_j \pmod{P_{A_0} p_j}$ if and only if $s - v \equiv \lambda_j v_j \pmod{P_{A_0}}$ and $s - v \equiv \lambda_j v_j \pmod{p_j}$. Observe that $\lambda_j v_j \equiv 0 \pmod{P_{A_0}}$. Hence, the first condition becomes $s - v \equiv 0 \pmod{P_{A_0}}$, which holds from the previous proof. Then, the second equivalence holds because $v = \sum_{i \in A_0} \lambda_i v_i \equiv 0 \pmod{p_j}$. Therefore, for each user $j \in A_0'$, the relation $s \equiv v_j \pmod{p_j}$ can be verified essentially for free given a UPoM proof between s and v , together with the share v_j and the public CRT coefficient λ_j . These observations forms the core idea behind our VWSS construction, enabling a single UPoM proof to simultaneously verify the shares of all parties.

3 Preliminaries

Notations. We denote security parameter by λ and negligible functions by $\text{negl}(\lambda)$. For positive integers m and n , we define $[n] = \{1, \dots, n\}$ and $[0, m] = \{0, 1, \dots, m-1\}$. For any two distributions X, Y over the finite universe Ω , the statistical distance between X and Y is defined as $\text{SD}(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. Finally, throughout the paper, we use ‘PPT’ short for probabilistic polynomial time.

⁴ The “global” CRT-coefficients are defined independently from any particular subset, as opposed to set-specific CRT-coefficients. For a system of n users, the global CRT-coefficients can be computed as $\lambda_i = \prod_{j=1, j \neq i}^n p_j \cdot \left(\prod_{j=1, j \neq i}^n p_j \right)^{-1} \pmod{p_i}$.

3.1 Groups of Unknown Order and Generic Group Model

We extensively use groups of unknown order (GUOs) in our constructions, where the group order cannot be computed efficiently. Suppose we are given a PPT algorithm GGen which, on input 1^λ , outputs a description of a finite abelian group $\mathbb{G}$, where we assume that, from the description, one can efficiently compute some upper bound on the order of the group (i.e., $\text{ord}(\mathbb{G}) < 2^U$ for $U \in \text{poly}(\lambda)$). We adopt the same security assumptions for GUOs as in [12].

Definition 1 (ORD). *The order ORD assumption holds for GGen if for any PPT adversary $\mathcal{A}$ it holds that*

$$\Pr \left[\begin{array}{l} g^\alpha = 1 \wedge \\ 1 \neq \alpha < 2^{\text{poly}(\lambda)} \end{array} \middle| \begin{array}{l} \mathbb{G} \leftarrow \text{GGen}(1^\lambda), \\ g \xleftarrow{\$} \mathbb{G}, \alpha \leftarrow \mathcal{A}(\mathbb{G}, g) \end{array} \right] \leq \text{negl}(\lambda)$$

Definition 2 (DLOG). *The Discrete Logarithm Relation DLOG assumption holds for GGen if for any PPT adversary $\mathcal{A}$ and positive integer $m \in \text{poly}(\lambda)$ it holds that*

$$\Pr \left[\begin{array}{l} \prod_{i=1}^m g_i^{\alpha_i} = 1 \wedge \\ 1 \neq \|\alpha\|_\infty < 2^{\text{poly}(\lambda)} \end{array} \middle| \begin{array}{l} \mathbb{G} \leftarrow \text{GGen}(1^\lambda), \\ g_1, \dots, g_m \xleftarrow{\$} \mathbb{G}, \\ (\alpha_1, \dots, \alpha_m) \leftarrow \mathcal{A}(\mathbb{G}, (g_1, \dots, g_m)) \end{array} \right] \leq \text{negl}(\lambda)$$

where $\alpha = (\alpha_1, \dots, \alpha_m)$ and $\|\alpha\|_\infty = \max_{i \in [m]} |\alpha_i|$.

Definition 3 (SI). *The subgroup indistinguishability (SI) assumption holds for GGen if for any PPT adversary $\mathcal{A}$, it holds that*

$$\Pr \left[b = b' \middle| \begin{array}{l} \mathbb{G} \leftarrow \text{GGen}(1^\lambda), g, h_0 \xleftarrow{\$} \mathbb{G}, h_1 \xleftarrow{\$} \langle g \rangle \\ b \xleftarrow{\$} \{0, 1\}, b' \leftarrow \mathcal{A}(\mathbb{G}, g, h_b) \end{array} \right] \leq \frac{1}{2} + \text{negl}(\lambda)$$

Generic Group Model for GUOs. In this paper, we use a generic group model (GGM) to argue for integral witness extraction. The GGM abstracts adversaries that cannot exploit any structural properties of group element encodings. Concretely, the group $\mathbb{G}$ is defined by a random injective map $\sigma : \mathbb{Z}_{|\mathbb{G}|} \rightarrow \{0, 1\}^\ell$ where $2^\ell \gg |\mathbb{G}|$ and group elements are given as $\sigma(0), \sigma(1), \dots, \sigma(|\mathbb{G}| - 1)$. A generic group adversary is a PPT algorithm that interacts with the group only through oracles providing random encodings and group operations on these encodings as follows:

- The oracle O_1 samples a random $r \in \mathbb{Z}_{|\mathbb{G}|}$ and returns $\sigma(r)$, which is appended to the list of encodings $\mathcal{L}$.
- When $\mathcal{L}$ has size q , the second oracle $O_2(i, j, \pm)$ takes two indices $i, j \in \{1, \dots, q\}$ and a sign bit, and returns $\sigma(x_i x_j^\pm)$, which is appended to $\mathcal{L}$.

It was shown in [1] that ORD and DLOG assumptions hold in the GGM.

Lemma 1. (*Element Representation [12,25]*) Let $\mathbb{G}$ be a generic group and $\mathcal{A}$ a generic algorithm making q_1 queries to O_1 and q_2 queries to O_2 . Let $\{g_1, \dots, g_m\}$ be the unique outputs of O_1 . There is an efficient algorithm $\mathcal{E}$ that given as input the transcript of $\mathcal{A}$'s interaction with the generic group oracles, produces for every element $u \in \mathbb{G}$ that $\mathcal{A}$ outputs, a tuple $(\alpha_1, \dots, \alpha_m) \in \mathbb{Z}^m$ such that $u = \prod_{i=1}^m g_i^{\alpha_i}$ and $\alpha_i \leq 2^{q_2}$ for all $i \in [m]$.

3.2 Commitments

Integer Commitment based on Groups of Unknown Order. We define an integer commitment scheme $\text{IC} = (\text{Setup}, \text{Commit}, \text{Open})$ over groups of unknown order with message space $\mathcal{M} = \mathbb{Z}$, following [12].

- $\text{IC.Setup}(1^\lambda)$. On input a security parameter λ in unary, the algorithm outputs $\text{IC.pp} = (\mathbb{G}, g, h)$, where $\mathbb{G} \leftarrow \text{GGen}(1^\lambda)$ and $g, h \xleftarrow{\$} \mathbb{G}$. We assume that IC.pp is an implicit input to the following algorithms.
- $\text{IC.Commit}(x)$. On input $x \in \mathbb{Z}$, sample $r \xleftarrow{\$} [0, S_r)$ with $S_r \geq 2^{U+\lambda}$, computes $c = g^x h^r$, set $d = (x, 1, r)$ and returns (c, d) .
- $\text{IC.Verify}(\text{IC.pp}, c, (x, (y, \ell, r)))$. The algorithm outputs 1 if and only if

$$\ell \neq 0 \wedge \ell \cdot x = y \wedge c^\ell = g^y h^r.$$

In the setting that does not desire a trusted setup, the public parameters additionally contain the random coins used in sampling group elements g, h . The integer commitment is correct, as an honest sender can always set $x = y$ and $\ell = 1$, so that the verification succeeds with probability 1. A commitment scheme satisfies binding and hiding properties, where binding prevents the committer from opening a commitment to a different message, and hiding ensures that the commitment reveals no information about the committed message. Following [12], the integer commitment scheme is binding under the DLOG assumption and statistically hiding with trusted setup or computationally hiding with transparent setup under the SI assumption.

Merkle-Tree-based Vector Commitment. We use a randomised variant of the Merkle-tree-based vector commitment [19]. Let $H_{\text{MT}} : \{0, 1\}^{4\lambda} \rightarrow \{0, 1\}^{2\lambda}$ be a collision-resistant hash function for internal nodes and let $H_{\text{leaf}} : \{0, 1\}^* \rightarrow \{0, 1\}^{2\lambda}$ be a random oracle for leaf commitments. The commitment scheme consists of algorithms $(\text{MT.Commit}, \text{MT.Open}, \text{MT.Verify})$, which are defined as follows:

- $(\text{com}_{\text{MT}}, \text{state}_{\text{MT}}) \leftarrow \text{MT.Commit}(\mathbf{V})$: On input a vector $\mathbf{V} = (v_1, \dots, v_M)$, where $v_e \in \{0, 1\}^*$, sample a list of random elements $(\rho_1, \dots, \rho_M) \xleftarrow{\$} \{0, 1\}^{M \times 2\lambda}$ and compute $\text{leaf}_e = H_{\text{leaf}}(v_e, \rho_e)$ for all $e \in [M]$. The algorithm then builds the binary Merkle tree from $(\text{leaf}_1, \dots, \text{leaf}_M)$ by computing the internal node with ordered child nodes z_L, z_R as $z = H_{\text{MT}}(z_L, z_R)$. The algorithm outputs the root $\text{com}_{\text{MT}} \in \{0, 1\}^{2\lambda}$ and the private opening state state_{MT} , containing $\mathbf{V}$, the salt $(\rho_e)_{e \in [M]}$ and the tree.

- $(v_e, \text{auth}_e) \leftarrow \text{MT.Open}(e, \text{state}_{\text{MT}})$: On input an index $e \in [M]$ and the private opening state state_{MT} , the algorithm outputs the message v_e and the corresponding authentication information $\text{auth}_e = (\rho_e, \text{path}_e)$, where path_e lists the sibling nodes on the path from position e to the root.
- $(0/1) \leftarrow \text{MT.Verify}(\text{com}_{\text{MT}}, v_e, \text{auth}_e, e)$: On input the Merkle root com_{MT} , message v_e , the corresponding authentication information $\text{auth}_e = (\rho_e, \text{path}_e)$, and the index $e \in [M]$, the algorithm recomputes $\text{leaf}_e = H_{\text{leaf}}(v_e, \rho_e)$ and use the leaf_e together with the sibling nodes in path_e to recompute com'_{MT} . It outputs 1 (accept) if $\text{com}'_{\text{MT}} = \text{com}_{\text{MT}}$, otherwise, outputs 0 (reject).

The Merkle-tree-based vector commitment is positional binding under the collision-resistance of H_{MT} and H_{leaf} . With fresh private salt for each leaf, the scheme is computationally hiding if H_{leaf} is modelled as a random oracle.

3.3 Verifiable Secret Sharing

A (non-interactive) verifiable secret sharing scheme (VSS) allows parties to verify that shares they received from the dealer are valid. We assume an authenticated broadcast channel and private authenticated channels from the dealer to the parties. The scheme consists of a tuple of PPT algorithms (**Setup**, **Share**, **Verify**, **Recon**) that are defined as follows.

- $\text{pp} \leftarrow \text{Setup}(1^\lambda)$: The setup algorithm outputs public parameters specifying the number of parties n , privacy threshold t , reconstruction threshold T , the collection of authorised sets $\mathfrak{A}$ and the collection of unauthorised sets $\bar{\mathfrak{A}}$ where $\mathfrak{A} \cap \bar{\mathfrak{A}} = \emptyset$.
- $((v_1, m_1) \dots, (v_n, m_n), \pi) \leftarrow \text{share}(\text{pp}, s)$: The sharing algorithm takes as input public parameters pp and the secret value s . It produces a public transcript π , a secret share v_i with auxiliary information m_i for every party $i \in [n]$. The dealer broadcasts π and privately sends (v_i, m_i) to party i .
- $(0/1) \leftarrow \text{Verify}(\text{pp}, \pi, i, v_i, m_i)$: Given the public parameters pp , the broadcast message π , the party index i , party's secret share v_i and the auxiliary information m_i , the algorithm outputs either 0 (invalid share) or 1 (valid share).
- $(\perp \vee s') \leftarrow \text{Recon}(\text{pp}, \{i, v_i, m_i\}_{i \in A}, \pi)$: The reconstruction algorithm takes as input public parameters pp , a set of shares $(v_i, m_i)_{i \in A}$. It outputs $\perp$ if $\text{Verify}(\text{pp}, \pi, i, v_i, m_i) = 0$ for some $i \in A$. Otherwise, it reconstruct the secret as s' .

Definition 4 (Security of VSS). *A non-interactive VSS scheme is secure against static corruptions if the following properties hold over the protocol coins of the honest parties and the adversary $\mathcal{A}$:*

- **Perfect Correctness.** *For every honestly generated pp and every authorised set $A \in \mathfrak{A}$, if $((v_i, m_i)_{i \in [n]}, \pi) \leftarrow \text{share}(\text{pp}, s)$, then*

$$\text{Recon}(\text{pp}, \{i, v_i, m_i\}_{i \in A}, \pi) = s.$$

- **Computational Secrecy.** For every PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the following experiment

- $\text{pp} \leftarrow \text{Setup}(1^\lambda)$,
- $(\bar{A}, s_0, s_1, \text{state}) \leftarrow \mathcal{A}_1(\text{pp})$,
- $b \xleftarrow{\$} \{0, 1\}$,
- $((v_i, m_i)_{i \in [n]}, \pi) \leftarrow \text{Share}(\text{pp}, s_b)$,
- $b' \leftarrow \mathcal{A}_2(\text{pp}, \text{state}, \pi, (v_j, m_j)_{j \in \bar{A}})$

We restrict to static adversaries, which choose $\bar{A} \in \bar{\mathfrak{A}}$ before generating shares. We say the scheme is computational secret if

$$\left| \Pr[b' = b] - \frac{1}{2} \right| \leq \text{negl}(\lambda)$$

- **Commitment (Consistency of Reconstruction).** For any PPT adversary $\mathcal{A}$, consider the experiment

- $\text{pp} \leftarrow \text{Setup}(1^\lambda)$,
- $(\pi, \{v_i, m_i\}_{i \in A}, \{v_i, m_i\}_{i \in \hat{A}}) \leftarrow \mathcal{A}(\text{pp})$,
- $x := \text{Recon}(\text{pp}, \{i, v_i, m_i\}_{i \in A}, \pi)$,
- $x' = \text{Recon}(\text{pp}, \{i, v_i, m_i\}_{i \in \hat{A}}, \pi)$,

where $A, \hat{A} \in \mathfrak{A}$ are two authorised sets. We say the scheme is consistent of reconstruction if

$$\Pr[x \neq \perp \wedge x' \neq \perp \wedge x \neq x'] \leq \text{negl}(\lambda).$$

3.4 Distributed Key Generation

A distributed key generation (DKG) protocol allows n parties to jointly generate a shared cryptographic key pair without relying on a centralised authority, and no single party possesses the complete secret key. At a high-level, a DKG scheme consists of three algorithms (Setup, Share, Verify, Recon), analogous to VSS defined in Section 3.3. In this paper, we design a DKG protocol which generates keys in the discrete logarithm setting. In this setting, given a cyclic group $\mathbb{G}_{\text{DL}}$ of prime order q , a DKG generates secret shares of a secret $x \in \mathbb{Z}_q$ and publishes $y = g_{\text{DL}}^x$ for a generator $g_{\text{DL}} \in \mathbb{G}_{\text{DL}}$.

Definition 5 (Security Definition of Distributed Key Generation). Let $\mathfrak{A}$ and $\bar{\mathfrak{A}}$ denote the collection of authorised set of parties and unauthorised parties, respectively. We say a distributed key generation protocol, denoted as Π_{DKG} , is secure if for every static PPT adversary $\mathcal{A}$ corrupting any unauthorised set $\bar{A} \in \bar{\mathfrak{A}}$, the following properties hold over the protocol coins of the honest parties and $\mathcal{A}$.

- **Correctness.**

- Agreement on the public key. All honest parties output the same public key.

- *Consistency of shares.* Each honest party j outputs a key share x_j such that, for every authorised set $A \in \mathfrak{A}$ whose members all output shares,

$$\text{Recon}(\{j, x_j\}_{j \in A}) = x \in \mathbb{Z}_q \quad \text{satisfies} \quad y = g_{\text{DL}}^x$$

- *Uniformity / bias-resistance.* The distribution of the recovered secret x over honest randomness is statistically close to uniform:

$$\text{SD}(x, U_{\mathbb{Z}_q}) \leq \text{negl}(\lambda).$$

- **Computational Secrecy.** There exists a PPT simulator Sim such that, on input the public parameters pp , an arbitrary $y^* \in \mathbb{G}_{\text{DL}}$, and the corruption set $\bar{A}$,

$$\text{Sim}(\text{pp}, y^*, \bar{A}) \stackrel{c}{\approx} \text{View}_{\bar{A}}(\Pi_{\text{DKG}} \mid y = y^*).$$

Equivalently, the adversary's view leaks at most y^* about x .

- **Commitment.** For every PPT adversary $\mathcal{A}$ that produces a transcript π together with two collections of shares corresponding to authorised sets $A, \hat{A} \in \mathfrak{A}$ that both pass the protocol's verification,

$$\Pr[\text{Recon}(\{j, x_j\}_{j \in A}, \pi) \neq \text{Recon}(\{j, x'_j\}_{j \in \hat{A}}, \pi) \neq \perp] \leq \text{negl}(\lambda).$$

Remark. We include the third correctness property since bias-resistance is a standard requirement for DKG protocols. Algorithm 5 lines 2–3 adopts a commit-then-open mechanism: each party first broadcasts $c_i = \text{Commit}(g_{\text{DL}}^{s_{i,0}})$ and only later opens it to reveal $y_i = g_{\text{DL}}^{s_{i,0}}$, so the binding property of **Commit** ensures that corrupted parties are bound to their contributions before honest parties' contributions are revealed.

3.5 Σ -Protocol and Zero Knowledge Proofs

At the core of our verifiable weighted secret sharing is Σ -protocol-based zero-knowledge proofs, which is defined below.

Definition 6 (Definition of Σ -Protocol based ZKP). Let $\mathcal{R}$ be an NP relation. For some $(x, w) \in \mathcal{R}$, we denote x as a public statement and w is a witness for x . A Σ -protocol Π is an interactive protocol between a prover P and a verifier V , where x is a common input to P, V and a w such that $(x, w) \in \mathcal{R}$ is a private input to P :

- P sends a commitment c_a .
- V sends a random challenge γ .
- P sends a response R_a and V decides to accept or reject based on the transcript $\text{tr} = (x, c_a, \gamma, R_a)$.

We say a protocol Π is a Σ -protocol for relation $\mathcal{R}$ if it satisfies the following security properties.

- **Correctness.** If P, V follow the protocol on input x and private w to P where $(x, w) \in \mathcal{R}$, the verifier always accepts.

- **Special Soundness.** For any x and any pair of accepting transcripts on input x , $(c_a, \gamma, R_a), (c_a, \gamma', R'_a)$ where $\gamma \neq \gamma'$, one can efficiently compute w such that $(x, w) \in \mathcal{R}$.
- **Special Honest Verifier Zero Knowledge (SHVZK).** There exists a polynomial-time simulator $\mathcal{S}$, which on input x and a random γ , outputs an accepting transcript of the form (c_a, γ, R_a) , with the same probability distribution as the transcript between the honest P, V on input x .

The following lemma plays a key role in proving the SHVZK of our unbounded proof-of-mod protocols.

Lemma 2 (Masking [6]). For any $C, B, I \in \mathbb{N}$ and fixed $x \in (-B, B)$, $\gamma \in (-C, C)$, the distribution of $X \xleftarrow{\$} [0, BCI]$ and $Y = \{m + \gamma x | m \xleftarrow{\$} [0, BCI]\}$ has statistical distance at most $1/I$.

Notably, rejection sampling and Gaussian noise allow us to use smaller masks, hence improve the communication complexity [18].

In our verifiable secret sharing protocol, we also assume black-box access to a range proof protocol $\Pi_{\text{RP}} = (\text{RP.Prove}, \text{RP.Verify})$ for the relation defined as

$$\mathcal{R}_{\text{RP}} = \{((c, B); m) : c = \text{IC.Commit}(m) \wedge m \in [0, B]\},$$

where (c, B) denotes the statement that includes the commitment c of the witness m and the range upper bound B on m . The range proof can be instantiated using the constructions in [6, 12], which are based on square decomposition and integer commitments. In our instantiation, we use the range proof of Couteau et al. [6], which satisfies 2-special soundness and honest verifier zero-knowledge as defined in Definition 6.

4 Unbounded Proof-of-Mod

In this section, we present two baseline protocols for constructing verifiable weighted secret sharing based on the Chinese Remainder Theorem. Both protocols are standard Σ -protocols that can be compiled into non-interactive zero-knowledge proofs via the Fiat–Shamir transform [11].

4.1 Unbounded Proof-of-Mod

We first show the *unbounded proof-of-mod* (UPoM) protocol where the prover holds $s \in [0, B)$ for an arbitrary integer $B \in \mathbb{N}$ and $v \equiv s \pmod{N}$.⁵ The modulus N (a prime or a square-free composite integer) is a common input to both prover and verifier.

⁵ This distinguishes our construction from the *bounded* proof-of-mod [24], which requires $s < q$, where q is the order of the commitment group. In contrast, our scheme imposes no restriction on B beyond being a positive integer, hence the term “unbounded” proof-of-mod.

Algorithm 1 Unbounded Proof-of-Mod (Π_{UPoM})

```

1: procedure witcom( $((a, S_a)_{a \in \{s, v\}}, B, E) \rightarrow ((c_a, d_a)_{a \in \{s, v\}})$ 
2:   Let  $s \in [0, B), v \in [0, E)$ ;
3:   Compute  $(c_a, (a, 1, r_a)) = \text{IC.Commit}(a)$  for  $r_a \xleftarrow{\$} [0, S_a)$  and  $a \in \{s, v\}$ ;
4:   Output  $(c_a, d_a)_{a \in \{s, v\}}$  where  $d_a = (a, 1, r_a)$ .
5: end procedure
6: procedure maskcom( $(S_a)_{a \in \{s, v\}}, B, E) \rightarrow ((\bar{c}_a, \bar{d}_a)_{a \in \{s, v\}}, \Delta)$ 
7:   Select  $\bar{s} \xleftarrow{\$} [0, BCI), \bar{v} \xleftarrow{\$} [0, ECI)$ ;
8:   Compute  $(\bar{c}_a, (\bar{a}, 1, \bar{r}_a)) = \text{IC.Commit}(\bar{a})$  for  $\bar{r}_a \xleftarrow{\$} [0, S_a CI)$  and  $a \in \{s, v\}$ ;
9:   Compute  $\Delta = \bar{s} - \bar{v} \pmod N$ ;
10:  Output  $((\bar{c}_a, \bar{d}_a)_{a \in \{s, v\}}, \Delta)$  where  $\bar{d}_a = (\bar{a}, 1, \bar{r}_a)$ .
11: end procedure
12: procedure ch( $((c_a, \bar{c}_a)_{a \in \{s, v\}}, \Delta) \rightarrow \gamma$ 
13:   Output  $\gamma \xleftarrow{\$} [0, C)$ .
14: end procedure
15: procedure resp( $((d_a, \bar{d}_a)_{a \in \{s, v\}}, \gamma) \rightarrow ((R_a, R_{r_a})_{a \in \{s, v\}})$ 
16:   Parse  $d_a = (a, 1, r_a)$  and  $\bar{d}_a = (\bar{a}, 1, \bar{r}_a)$  for  $a \in \{s, v\}$ ;
17:   Compute  $R_a = \bar{a} + a \cdot \gamma$  and  $R_{r_a} = \bar{r}_a + r_a \cdot \gamma$  for  $a \in \{s, v\}$ ;
18:   Output  $(R_a, R_{r_a})_{a \in \{s, v\}}$ .
19: end procedure
20: procedure ver( $((c_a, \bar{c}_a, R_a, R_{r_a})_{a \in \{s, v\}}, \Delta, \gamma, B, E) \rightarrow 0/1$ 
21:   Check if  $g^{R_a} h^{R_{r_a}} = \bar{c}_a \cdot c_a^\gamma$  for  $a \in \{s, v\}$ ;
22:    $\Delta \stackrel{?}{=} R_s - R_v \pmod N$  for  $R_s \in [0, BC(I+1))$  and  $R_v \in [0, EC(I+1))$ ;
23:   Output 1 (accept) if all checks passed. Otherwise, output 0 (reject).
24: end procedure

```

Algorithm 1 (Π_{UPoM}) is a Σ -protocol for the unbounded proof-of-mod relation defined as

$$\mathcal{R}_{\text{UPoM}} = \{((c_s, c_v, N); (s, v)) : v \equiv s \pmod N\}.$$

We assume that all sub-algorithms (witcom, maskcom, ch, resp, ver) of Π_{UPoM} implicitly take as input a set of public parameters $\Pi_{\text{UPoM}}.\text{pp} = (\lambda, \text{IC.pp}, U, I, C)$, where λ is the security parameter, IC.pp is the public parameters for the integer commitment scheme, U is the upper bound on the bit length of the group order, and $C = I = 2^\lambda$.

Theorem 1. *The unbounded proof-of-mod shown in Algorithm 1 is 2-special sound if IC is binding, ORD and DLOG assumptions are hard in the generic group model. Let $N = p_1 \cdot p_2 \cdots p_m$ where $p_1 < p_2 < \cdots < p_m$, the soundness error ϵ is defined as follows:*

$$\epsilon \leq \begin{cases} \frac{1}{C} & \text{if } p_1 \geq C \\ \sum_{i=1}^{m'} \left(\frac{1}{p_i} + \frac{1}{4C} \right) & \text{if } p_{m'} < C \leq p_{m'+1} \text{ for } m' \leq m. \end{cases}$$

Proof. Let $(R_s, R_{r_s}, R_v, R_{r_v})$ and $(R'_s, R'_{r_s}, R'_v, R'_{r_v})$ be two responses of accepting transcripts generated by $\mathcal{A}$ with respect to challenges $\gamma \neq \gamma'$. Since the commitments $(c_s, c_v, \bar{c}_s, \bar{c}_v)$ are consistent in both transcripts, for $a \in \{s, v\}$, we have

$$g^{R_a} h^{R_{r_a}} c_a^{-\gamma} = g^{R'_a} h^{R'_{r_a}} c_a^{-\gamma'} = \bar{c}_a. \quad (2)$$

It follows that

$$g^{R_a - R'_a} h^{R_{r_a} - R'_{r_a}} c_a^{-(\gamma - \gamma')} = 1.$$

Let $R_a^* = R_a - R'_a$, $R_{r_a}^* = R_{r_a} - R'_{r_a}$ and $\gamma^* = \gamma - \gamma'$, we have

$$g^{R_a^*} h^{R_{r_a}^*} c_a^{-\gamma^*} = 1 \implies g^{R_a^*} h^{R_{r_a}^*} = c_a^{\gamma^*}.$$

Thus, $(x_a, (R_a^*, \gamma^*, R_{r_a}^*))$ is a valid relaxed opening of c_a , where $x_a = \frac{R_a^*}{\gamma^*}$. And by the binding property of the integer commitment scheme, the opening is unique.

If the adversary is generic, then by Lemma 1, there exists an integer representation of $c_a = g^{\bar{R}_a} h^{\bar{R}_{r_a}} \prod_j u_j^{\bar{b}_j}$ and $u_j \neq g, h$ for all j , where $a \in \{s, v\}$. The integers $\bar{R}_a, \bar{R}_{r_a}, \bar{b}_j$ can be efficiently computed from $\mathcal{A}$'s queries to the generic group oracle. It follows that

$$c_a^{\gamma^*} = g^{R_a^*} h^{R_{r_a}^*} = g^{\gamma^* \cdot \bar{R}_a} h^{\gamma^* \cdot \bar{R}_{r_a}} \prod_j u_j^{\gamma^* \cdot \bar{b}_j}.$$

By the DLOG assumption in GGM, with overwhelming probability, we have $R_a^* = \gamma^* \cdot \bar{R}_a$, $R_{r_a}^* = \gamma^* \cdot \bar{R}_{r_a}$ and $\bar{b}_j = 0$ for all j .

Suppose by contradiction that $c_a^* = g^{\frac{R_a^*}{\gamma^*}} h^{\frac{R_{r_a}^*}{\gamma^*}} \neq c_a$, since $(c_a^*)^{\gamma^*} = g^{R_a^*} h^{R_{r_a}^*} = c_a^{\gamma^*}$, we have $(c_a^* \cdot c_a^{-1})^{\gamma^*} = 1$. Hence, $\frac{c_a^*}{c_a}$ is a non-trivial element of order dividing γ^* . As γ^* is efficiently computable, this would contradict the ORD assumption in the GGM. Hence, $c_a^* = c_a$ with overwhelming probability, and $\frac{R_a^*}{\gamma^*}, \frac{R_{r_a}^*}{\gamma^*} \in \mathbb{Z}$ for $a \in \{s, v\}$.

Let us write $a = \frac{R_a^*}{\gamma^*} \in \mathbb{Z}$ for $a \in \{s, v\}$ and the relation enforcement $\Delta = R_s - R_v \equiv R'_s - R'_v \pmod{N}$. Since Δ is fixed, we have $R_s - R_v \equiv R'_s - R'_v \pmod{N}$. Hence,

$$R_s - R'_s - (R_v - R'_v) \equiv 0 \pmod{N} \implies R_s^* - R_v^* \equiv 0 \pmod{N}.$$

Recall that $R_a^* = a \cdot \gamma^*$ for $a \in \{s, v\}$, we have

$$s \cdot \gamma^* - v \cdot \gamma^* \equiv 0 \pmod{N}. \quad (3)$$

Let $N = \prod_{i=1}^m p_i$ where $p_1 < p_2 < \dots < p_m$. If $C \leq p_1$, then $\gamma^* \in (-C, C)$ must have $\gcd(\gamma^*, N) = 1$ except that $\gamma^* \equiv 0 \pmod{N} \implies \gamma \equiv \gamma' \pmod{N}$. Hence, we can cancel out γ^* from both side of Equation 3 and get the relation

$$s - v \equiv 0 \pmod{N}$$

with probability $1 - \frac{1}{C}$.

On the other hand, if $p_1 < p_2 < \dots < p_{m'} < C \leq p_{m'+1} < \dots < p_m$ for any $m' \leq m$, then $\gcd(\gamma^*, N) \neq 1 \iff \gamma = \gamma' \pmod{p_i}$ for some $p_i \in \{p_1, \dots, p_{m'}\}$. Let $C = p_i q_i + r_i$ for $0 \leq r_i < p_i$, it follows that

$$\begin{aligned} \Pr[\gamma = \gamma' \pmod{p_i}] &= \frac{r_i(q_i + 1)^2 + (p_i - r_i)q_i^2}{C^2} \\ &\leq \frac{1}{p_i} + \frac{p_i}{4C^2} < \frac{1}{p_i} + \frac{1}{4C}. \end{aligned}$$

Taking the union bound, we have

$$\Pr[\gcd(\gamma^*, N) \neq 1] < \sum_{i=1}^{m'} \left(\frac{1}{p_i} + \frac{1}{4C} \right)$$

Hence, with probability at least $1 - \sum_{i=1}^{m'} \left(\frac{1}{p_i} + \frac{1}{4C} \right)$, we can cancel γ^* from both sides of Equation 3 and get the desired proof-of-mod relation. $\square$

Theorem 2. *Algorithm 1 is special honest verifier zero-knowledge if IC is hiding.*

Proof. We modify the distribution of valid transcripts in four indistinguishable steps such that the last game does not require a witness as input. The simulator defined by game 4 thus fulfils the requirements of the zero-knowledge definition.

Game 1. Outputs an unmodified transcript from an interaction of an honest verifier and prover from the protocol definition:

- Witness Commitment: $(c_a, (a, 1, r_a)) = \text{IC.Commit}(a)$ for $a \in \{s, v\}$.
- Mask Commitment: $(\bar{c}_a, (\bar{a}, 0, \bar{r}_a)) = \text{IC.Commit}(\bar{a})$ for $a \in \{s, v\}$,
where $\bar{s} \xleftarrow{\$} [0, BCI], \bar{v} \xleftarrow{\$} [0, NCI]$.
- Relation enforcement $\Delta = \bar{s} - \bar{v} \pmod{N}$.
- Challenge: $\gamma \xleftarrow{\$} [0, C]$.
- Response: $R_a = \bar{a} + a \cdot \gamma$ and $R_{r_a} = \bar{r}_a + r_a \cdot \gamma$ for $a \in \{s, v\}$.

Game 2. Rewrites the mask commitments $\bar{c}_s, \bar{c}_v$ and relation enforcement Δ as in the verification check:

- Witness Commitment: $(c_a, (a, 1, r_a)) = \text{IC.Commit}(a)$ for $a \in \{s, v\}$.
- Set $\bar{s} \xleftarrow{\$} [0, BCI]$ and $\bar{v} \xleftarrow{\$} [NCI]$.
- Responses: $R_a = \bar{a} + a \cdot \gamma$ and $R_{r_a} = \bar{r}_a + r_a \cdot \gamma$ for $a \in \{s, v\}$.
- Mask Commitment: $\bar{c}_a = g^{R_a} h^{R_{r_a}} c_a^{-\gamma}$ for $a \in \{s, v\}$.
- Relation enforcement $\Delta = R_s - R_v \pmod{N}$.

for given $\gamma \in [0, C], s \in [0, B], v \in [0, N]$.

Game 1 and Game 2 are identically distributed due to the correctness of the protocol. That is, $g^{R_a} h^{R_{r_a}} = \bar{c}_a c_a^\gamma = g^{R_a} h^{R_{r_a}} c_a^{-\gamma} c_a^\gamma = g^{R_a} h^{R_{r_a}}$ for $a \in \{s, v\}$ and $\Delta = R_s - R_v = \bar{s} - \bar{v} + \gamma(s - v) = \bar{s} - \bar{v} \bmod N$ as $s \equiv v \bmod N$.

Game 3. Remove the dependency on witnesses in the responses.

- Witness Commitment: $(c_a, (a, 1, r_a)) = \text{IC.Commit}(a)$ for $a \in \{s, v\}$.
- Responses $R_s \xleftarrow{\$} [0, BCI]$, $R_v \xleftarrow{\$} [0, NCI]$, and $R_{r_a} \xleftarrow{\$} [S_a CI]$ for $a \in \{s, v\}$.
- Relation enforcement $\Delta = R_s - R_v \bmod N$.
- Mask Commitment: $\bar{c}_a = g^{R_a} h^{R_{r_a}} c_a^{-\gamma}$ for $a \in \{s, v\}$.

for given $\gamma \in [0, C]$, $s \in [0, B]$, $v \in [0, N]$.

According to Lemma 2, Game 2 and Game 3 have statistical distance at most $4/I$. Also note that the distribution of Δ is unchanged as R_a and $\bar{a}$ are all uniformly random from the same range for $a \in \{s, v\}$.

Game 4. Remove the dependency on witnesses in the commitments.

- Witness Commitment: $(c_a, (0, 1, r_a)) = \text{IC.Commit}(0)$ for $a \in \{s, v\}$.
- Responses $R_s \xleftarrow{\$} [0, BCI]$, $R_v \xleftarrow{\$} [0, NCI]$, and $R_{r_a} \xleftarrow{\$} [S_a CI]$ for $a \in \{s, v\}$.
- Relation enforcement $\Delta = R_s - R_v \bmod N$.
- Mask Commitment: $\bar{c}_a = g^{R_a} h^{R_{r_a}} c_a^{-\gamma}$ for $a \in \{s, v\}$.

for given $\gamma \xleftarrow{\$} [0, C]$.

Game 3 and Game 4 are indistinguishable under the hiding property of the commitment IC. $\square$

Remark 1. We assume that $\bar{s}$ is sampled from an interval exponentially larger than N . By Claim 1, $(\bar{s} - \bar{v}) \bmod N$ is statistically close to uniformly distributed over $\mathbb{Z}_N$, since $\bar{s}$ and $\bar{v}$ are sampled independently.

Claim 1. Let $M < S$ be arbitrary integers and $U_s \xleftarrow{\$} [0, S)$, then we have

$$\text{SD}(U_s \bmod M, U_M) \leq \frac{M}{4S},$$

where SD denotes the statistical distance and $U_M \xleftarrow{\$} \mathbb{Z}_M$. Furthermore, let X, Y be independent random variables over $\mathbb{Z}_M$, then

$$\text{SD}(X - Y \bmod M, U_M) \leq \text{SD}(X, U_M).$$

The proof of Claim 1 is standard and analogous to that in [13, Claim 1].

Proof. Let $S = M \cdot q + r$ for $0 \leq r < M$. Then the distribution of $U_s \bmod M$ where $U_s \xleftarrow{\$} [0, S-1]$ is $\frac{q+1}{S}$ (for r residues) or $\frac{q}{S}$ (for $M-r$ residues). Let $U_M \xleftarrow{\$} \mathbb{Z}_M$, by definition, we have the statistical distance between $U_s \bmod M$ and $U_M \xleftarrow{\$} \mathbb{Z}_M$ is

$$\begin{aligned} \text{SD}(U_s \bmod M, U_M) &= \frac{1}{2} \sum_{a=0}^{M-1} \left| \Pr[U_s \bmod M = a] - \frac{1}{M} \right| \\ &= \frac{1}{2} \left(r \left(\frac{q+1}{S} - \frac{1}{M} \right) + (M-r) \left(\frac{1}{M} - \frac{q}{S} \right) \right) \\ &= \frac{r(M-r)}{MS}. \end{aligned}$$

Since $r(M-r) = \frac{M^2}{4} - \left(r - \frac{M}{2}\right)^2 \leq \frac{M^2}{4}$, we have $\text{SD}(U_s \bmod M, U_M) \leq \frac{M}{4S}$.

For the second part, let $Z = X - Y \bmod M$. For every $z \in \mathbb{Z}_M$, we have

$$\Pr[Z = z] = \sum_{y \in \mathbb{Z}_M} \Pr[Y = y] \Pr[X = z + y \bmod M]$$

Subtracting both sides by $\frac{1}{M}$ and by triangle inequality and $\Pr[Y = y] \geq 0$, we have

$$\left| \Pr[Z = z] - \frac{1}{M} \right| \leq \sum_{y \in \mathbb{Z}_M} \Pr[Y = y] \left| \left(\Pr[X = z + y \bmod M] - \frac{1}{M} \right) \right|.$$

Sum both sides over $z \in \mathbb{Z}_M$, we get

$$\sum_{z \in \mathbb{Z}_M} \left| \Pr[Z = z] - \frac{1}{M} \right| \leq \sum_{y \in \mathbb{Z}_M} \Pr[Y = y] \sum_{z \in \mathbb{Z}_M} \left| \left(\Pr[X = z + y \bmod M] - \frac{1}{M} \right) \right|$$

For fixed y , $z + y \bmod M$ over all $z \in \mathbb{Z}_M$ is a shift over all residues mod M , so we can rename $x = z + y \bmod M$ to simplify the inner sum to $\sum_{x \in \mathbb{Z}_M} \left| \Pr[X = x] - \frac{1}{M} \right|$ and since $\sum_{y \in \mathbb{Z}_M} \Pr[Y = y] = 1$, it follows that

$$\sum_{z \in \mathbb{Z}_M} \left| \Pr[Z = z] - \frac{1}{M} \right| \leq \sum_{x \in \mathbb{Z}_M} \left| \Pr[X = x] - \frac{1}{M} \right|.$$

It follows that $\text{SD}(X - Y \bmod M, U_M) \leq \text{SD}(X, U_M)$. $\square$

4.2 Boosting Soundness for Unbounded Proof-of-Mod

When N contains small prime factors, the UPoM (Algorithm 1) exhibits poor soundness. To improve the soundness, we let the prover commit to s, v, k (witnesses) and send the commitments c_s, c_v, c_k to the verifier. Algorithm 2 (Π_{UPoMS})

Algorithm 2 UPoM with Boosted Soundness (Π_{UPoMS})

```

1: procedure witcom( $((a, S_a)_{a \in \{s,k,v\}}, B, D, E) \rightarrow ((c_a, d_a)_{a \in \{s,k,v\}})$ 
2:   Input  $s \in [0, B)$ ,  $k \in [0, D)$ ,  $v \in [0, E)$ ;
3:   Compute  $(c_a, (a, 1, r_a)) = \text{IC.Commit}(a)$  for  $r_a \xleftarrow{\$} [0, S_a)$  and  $a \in \{s, k, v\}$ ;
4:   Output  $(c_a, d_a)_{a \in \{s,k,v\}}$  where  $d_a = (a, 1, r_a)$ .
5: end procedure
6: procedure maskcom( $((S_a)_{a \in \{s,k,v\}}, B, D, E) \rightarrow ((\bar{c}_a, \bar{d}_a)_{a \in \{s,k,v\}}, \Delta)$ 
7:   Select  $\bar{s} \xleftarrow{\$} [0, BCI)$ ,  $\bar{k} \xleftarrow{\$} [0, DCI)$  and  $\bar{v} \xleftarrow{\$} [0, ECI)$ ;
8:   Compute  $(\bar{c}_a, (\bar{a}, 1, \bar{r}_a)) = \text{IC.Commit}(\bar{a})$  for  $\bar{r}_a \xleftarrow{\$} [0, S_a CI)$  and  $a \in \{s, k, v\}$ ;
9:   Compute  $\Delta = \bar{s} - \bar{k}N - \bar{v}$ ;
10:  Output  $((\bar{c}_a, \bar{d}_a)_{a \in \{s,k,v\}}, \Delta)$  where  $\bar{d}_a = (\bar{a}, 1, \bar{r}_a)$ .
11: end procedure
12: procedure ch( $((c_a, \bar{c}_a)_{a \in \{s,k,v\}}, \Delta) \rightarrow \gamma$ 
13:   Output  $\gamma \xleftarrow{\$} [0, C)$ .
14: end procedure
15: procedure resp( $((d_a, \bar{d}_a)_{a \in \{s,k,v\}}, \gamma) \rightarrow ((R_a, R_{r_a})_{a \in \{s,k,v\}})$ 
16:   Parse  $d_a = (a, 1, r_a)$  and  $\bar{d}_a = (\bar{a}, 1, \bar{r}_a)$  for  $a \in \{s, k, v\}$ ;
17:   Compute  $R_a = \bar{a} + a \cdot \gamma$  and  $R_{r_a} = \bar{r}_a + r_a \cdot \gamma$  for  $a \in \{s, k, v\}$ ;
18:   Output  $(R_a, R_{r_a})_{a \in \{s,k,v\}}$ .
19: end procedure
20: procedure ver( $((c_a, \bar{c}_a, R_a, R_{r_a})_{a \in \{s,k,v\}}, \Delta, \gamma, B, D, E) \rightarrow 0/1$ 
21:    $g^{R_a} h^{R_{r_a}} \stackrel{?}{=} \bar{c}_a \cdot c_a^\gamma$  for  $a \in \{s, k, v\}$ ;
22:    $\Delta \stackrel{?}{=} R_s - R_v - NR_k$ ;
23:   Check if  $R_s \in [0, BC(I+1))$ ,  $R_k \in [0, DC(I+1))$ ,  $R_v \in [0, EC(I+1))$ ;
24:   Output 1 (accept) if all checks passed. Otherwise, output 0 (reject).
25: end procedure

```

is a Σ -protocol with boosted soundness for the unbounded proof-of-mod relation that is defined as

$$\mathcal{R}_{\text{UPoMS}} = \{((c_s, c_v, c_k, N); (s, v, k)) : s = v + kN\}.$$

In the protocol, both the prover and the verifier are assumed to have access to common public parameters, which are defined the same as in Section 4.1.

Theorem 3. *Algorithm 2 is a 2-special sound protocol for unbounded proof-of-mod relation $\mathcal{R}_{\text{UPoMS}}$ if IC is binding, DLOG and ORD assumptions are hard in the generic group model, with soundness error $\frac{1}{C}$.*

Proof. Assume there are two accepting transcripts with *distinct* challenges $\gamma \neq \gamma'$ and responses $(R_s, R_{r_s}, R_k, R_{r_k}, R_v, R_{r_v})$ and $(R'_s, R'_{r_s}, R_k, R'_{r_k}, R'_v, R'_{r_v})$ generated by $\mathcal{A}$. We can follow the same steps as the proof of Theorem 1 to extract valid relaxed openings of c_a as $(x_a, (R_a^*, \gamma^*, R_{r_a}^*))$ where $x_a = \frac{R_a^*}{\gamma^*}$ and $R_a^* = R_a - R'_a$, $R_{r_a}^* = R_{r_a} - R'_{r_a}$, $\gamma^* = \gamma - \gamma'$ for $a \in \{s, k, v\}$. If $\mathcal{A}$ is generic, it follows that $\frac{R_a^*}{\gamma^*} \in \mathbb{Z}$. Let $a = \frac{R_a^*}{\gamma^*} \in \mathbb{Z}$, given the fixed relation enforcement

$\Delta = R_s - R_v - NR_k = R'_s - R'_v - NR'_k$, we have $R_s^* = R_v^* + NR_k^*$. As $R_a^* = a \cdot \gamma^*$, we obtain

$$s \cdot \gamma^* = v \cdot \gamma^* + kN \cdot \gamma^*.$$

If $\gamma^* \neq 0$, then we can cancel γ^* from both side of the equation and get

$$s = v + kN$$

with probability $1 - \frac{1}{C}$. $\square$

Theorem 4. *Algorithm 2 is special honest verifier zero-knowledge if IC is hiding.*

Proof. The proof is analogous to that of Theorem 2 and we omit the details.

5 Verifiable Weighted Secret Sharing

The setup algorithm of our verifiable weighted secret sharing (VWSS) scheme generates the following parameters that are implicit inputs to the dealer and all users.

- Unbounded proof-of-mod parameters $\Pi_{\text{UPoM}}.\text{pp}$ defined as in Section 4.1,
- $H_{\text{FS}} : \{0, 1\}^* \rightarrow \{0, 1\}^{2\lambda}$: a hash function used for Fiat-Shamir transform that is modelled as a random oracle,
- $H_{\text{leaf}} : \{0, 1\}^* \rightarrow \{0, 1\}^{2\lambda}$: a hash function used for leaf node commitment that is modelled as a random oracle,
- $H_{\text{MT}} : \{0, 1\}^* \rightarrow \{0, 1\}^{2\lambda}$: a collision-resistant hash function used for computing internal nodes of the Merkle-tree-based vector commitment,
- n : the number of users,
- $p_0, p_1, \dots, p_n$: distinct odd primes and $P = \prod_{i=1}^n p_i$,
- $T \in \mathbb{N}$: the reconstruction threshold,
- $t \in \mathbb{N}$: the privacy threshold,
- $L \in \mathbb{N}$: the CRT secret sharing lifting parameter where $2^{t+\lambda} \leq L \leq 2^{T-\lambda}$,
- A_0 : the set with sufficient weights to recover the secret and $P_{A_0} = \prod_{i \in A_0} p_i$,
- $A'_0 = [n] \setminus A_0$: the users that are not in A_0 ,
- $\lambda_i \in (-\frac{P}{2}, \frac{P}{2})$ for $i \in [n]$: the *global* CRT-coefficients, where $\lambda_i = \prod_{j=1, j \neq i}^n p_j \cdot \left(\prod_{j=1, j \neq i}^n p_j \right)^{-1} \bmod p_i$.

We present the VWSS secret sharing in Algorithm 3 and the corresponding verification in Algorithm 4. The reconstruction procedure is identical to that of the standard CRT-based secret sharing scheme described in [13], and is therefore omitted.

Following the discussion in Section 2, the VWSS protocol first partitions the n users to a predefined authorised set A_0 and $A'_0 = [n] \setminus A_0$. Let $s \in [0, B)$ be the lifted CRT secret and $v_i = s \bmod p_i$ be secret shares. Define $v = \sum_{i \in A_0} \lambda_i v_i$

for global CRT coefficients λ_i . In the beginning, the dealer commits to s and shares $\{v_i\}_{i \in A_0}$, then generates a single UPoM proof establishing that

$$s \equiv v \pmod{P_{A_0}}.$$

This proof shows that all shares in A_0 are generated correctly. Then, for each user in $j \in A'_0$, if $p_j \geq 2^\lambda$, the dealer computes a separate relation enforcement $\Delta_j = \bar{s} - \bar{v} \pmod{p_j}$ otherwise, the dealer computes and commits to k_j such that $s = v + \lambda_j v_j + k_j p_j$ and the corresponding relation enforcement $\Delta_j = \bar{s} - \bar{v} - \bar{k}_j p_j$. This establishes that

$$s \equiv v + \lambda_j v_j \pmod{p_j}$$

through UPoM protocol.

In our protocol, the choice of A_0 creates a trade-off between broadcast complexity, proof generation time and private message complexity.

1. **Method 1.** We include all small primes (those less than λ bits) in A_0 . As a result, A'_0 contains only parties with large weights. In this setting, the private message size is minimised: for $i \in A_0$, it contains only (v_i, r_{v_i}) , while for $j \in A'_0$, it contains $(v_j, \Delta_j, \text{auth}_j)$. This comes at the cost of a larger broadcast message but faster proof generation.
2. **Method 2.** We include the largest primes in A_0 . In this case, A_0 contains the fewest possible parties, leading to a smaller broadcast message. However, since A'_0 now contains parties with small primes, the dealer must additionally generate k_j using the UPoM with soundness boost for these parties. Consequently, the private message for low-weight users in A'_0 is larger, while the private message to other parties (i.e., parties in A_0 and high-weight parties in A'_0) remain unchanged. Due to the additional computation required for partial UPoM proofs corresponding to users $j \in A'_0$ with small weights, the proof generation time is longer with this method.

Theorem 5. *Let H_{FS} and H_{leaf} be modelled as independent random oracles, and H_{MT} be a collision-resistant hash function. If the integer commitment is binding and hiding, the unbounded proof-of-mod (with soundness boost) and range proof are complete, special sound, and special honest verifier zero-knowledge in the random oracle model, then the verifiable weighted secret sharing protocol (Algorithm 3 and 4) achieves correctness, secrecy, and commitment properties as defined in Definition 4.*

Proof. Let $\mathfrak{A}$ be the collection of authorised sets and $\bar{\mathfrak{A}}$ be the collection of unauthorised sets.

Correctness. An honest sharing satisfies $s \in [0, B)$ and $s \equiv v_i \pmod{p_i}$ for every party. With $v = \sum_{A_0} \lambda_i v_i$, the global CRT coefficients guarantee that $s \equiv v \pmod{P_{A_0}}$ and $s \equiv v + \lambda_j v_j \pmod{p_j}$ for $j \in A'$. Following the completeness of UPoM protocols, the range proof, and the correctness of the integer and Merkle-tree-based vector commitment openings, every honest share is accepted. Furthermore, for any authorised set $A \in \mathfrak{A}$, since $P_A = \prod_{i \in A} p_i \geq B$, by the

Algorithm 3 VWSS: Secret Sharing

- 1: Lifted CRT-secret $s \in [0, B)$ and CRT secret share $v_i = s \bmod p_i$ for user $i \in [n]$.
 - 2: Compute $(c_s, (s, 0, r_s)) = \text{IC.Commit}(s)$ and $(c_{v_i}, (v_i, 0, r_{v_i})) = \text{IC.Commit}(v_i)$ for $i \in A_0$, where $r_s, r_{v_i} \in [0, S)$ for $S = 2^{U+\lambda}$.
 - 3: Define $v = \sum_{i \in A_0} \lambda_i v_i$ for $v \in (-E, E)$ where $E = \frac{P}{2} \sum_{i \in A_0} p_i$, $r_v = \sum_{i \in A_0} \lambda_i r_{v_i}$ for $r_v \in (-S_v, S_v)$, $S_v = \frac{P}{2} S |A_0|$, and $B' = \max(B, P_{A_0} \cdot 2^t)$.
 - 4: **if** $\forall i \in A_0, p_i \geq C$ **then**
 - 5: Get $((\bar{c}_a, \bar{d}_a)_{a \in \{s, v\}}, \Delta_{A_0}) \leftarrow \Pi_{\text{UPoM}}.\text{maskcom}(S, S_v, B', E)$.
 - 6: **else**
 - 7: Compute $k \in (-D, D)$ where $s = v + kP_{A_0}$ and $D = \left\lceil \frac{B+E}{P_{A_0}} \right\rceil$.
 - 8: Compute $(c_k, (k, 1, r_k)) = \text{IC.Commit}(k)$.
 - 9: Get $((\bar{c}_a, \bar{d}_a)_{a \in \{s, k, v\}}, \Delta_{A_0}) \leftarrow \Pi_{\text{UPoMS}}.\text{maskcom}(S, S_k, S_v, B', D, E)$.
 - 10: **end if**
 - 11: **for** $j \in A'_0$ **do**
 - 12: Compute $v_j = s \bmod p_j$.
 - 13: **if** $p_j \geq C$ **then**
 - 14: Compute $\Delta_j = \bar{s} - \bar{v} \bmod p_j$ and define $\text{tr}_j = (\Delta_j, v_j)$.
 - 15: **else**
 - 16: Compute $k_j \in (-D_j, D_j)$ where $s = v + \lambda_j v_j + k_j p_j$ and $D_j = \left\lceil \frac{B+E}{p_j} + \frac{P}{2} \right\rceil$.
 - 17: Commit to k_j as $(c_{k_j}, (k_j, 1, r_{k_j})) = \text{IC.Commit}(k_j)$ where $r_{k_j} \in [0, S)$.
 - 18: Compute $\bar{k}_j \xleftarrow{\$} [0, D_j CI)$ and $\bar{k}_j$ as $(\bar{c}_{k_j}, (\bar{k}_j, 1, \bar{r}_{k_j}))$ where $\bar{r}_{k_j} \in [0, SCI)$.
 - 19: Compute $\Delta_j = \bar{s} - \bar{v} - \bar{k}_j p_j$ and define $\text{tr}_j = (c_{k_j}, \bar{c}_{k_j}, \Delta_j, v_j)$.
 - 20: **end if**
 - 21: **end for**
 - 22: Compute the Merkle tree commitment $(\text{com}_{\text{MT}}, \text{state}_{\text{MT}}) \leftarrow \text{MT.Commit}((\text{tr}_j)_{j \in A'_0})$ and the range proof on s as $\pi_{\text{RP}} \leftarrow \text{RP.Prove}((c_s, B), s)$.
 - 23: **if** $\forall i \in A_0, p_i \geq C$ **then**
 - 24: $\gamma \leftarrow H_{\text{FS}}(c_s, \{c_{v_i}\}_{i \in A_0}, \bar{c}_s, \bar{c}_v, \Delta_{A_0}, \text{com}_{\text{MT}})$.
 - 25: Get $((R_a, R_{r_a})_{a \in \{s, v\}}) \leftarrow \Pi_{\text{UPoM}}.\text{resp}((d_a, \bar{d}_a)_{a \in \{s, v\}}, \gamma)$.
 - 26: Broadcasts $(c_s, \{c_{v_i}\}_{i \in A_0}, R_s, R_{r_s}, R_v, R_{r_v}, \text{com}_{\text{MT}}, \gamma, \pi_{\text{RP}})$.
 - 27: **else**
 - 28: $\gamma \leftarrow H_{\text{FS}}(c_s, \{c_{v_i}\}_{i \in A}, c_k, \bar{c}_s, \bar{c}_k, \bar{c}_v, \Delta_{A_0}, \text{com}_{\text{MT}})$.
 - 29: Get $(R_a, R_{r_a})_{a \in \{s, k, v\}} \leftarrow \Pi_{\text{UPoMS}}.\text{resp}((d_a, \bar{d}_a)_{a \in \{s, k, v\}}, \gamma)$.
 - 30: Broadcast $(c_s, c_k, \{c_{v_i}\}_{i \in A_0}, R_s, R_{r_s}, R_k, R_{r_k}, R_v, R_{r_v}, \text{com}_{\text{MT}}, \gamma, \pi_{\text{RP}})$.
 - 31: **end if**
 - 32: For $j \in A'_0$, if $p_j < C$, then $R_{k_j} = \bar{k}_j + k_j \cdot \gamma$ and $R_{r_{k_j}} = \bar{r}_{k_j} + r_{k_j} \cdot \gamma$.
 - 33: For $i \in A_0$, the dealer sends (v_i, r_{v_i}) through the private channel.
 - 34: For $j \in A'_0$, if $p_j \geq C$ the dealer sends (v_j, auth_j) to j through the private channel, where $(\cdot, \text{auth}_j) \leftarrow \text{MT.Open}(j, \text{state}_{\text{MT}})$ is the Merkle tree opening information with respect to index j . If $p_j < C$, the dealer sends $(v_j, c_{k_j}, R_{k_j}, R_{r_{k_j}}, \text{auth}_j)$ through the private channel.
-

Algorithm 4 VWSS: Verification

```

1: for  $i \in [n]$  do
2:   Compute  $\bar{c}'_s = g^{R_s} h^{R_{rs}} \cdot c_s^{-\gamma}$  and  $\bar{c}'_v = g^{R_v} h^{R_{rv}} \cdot \left(\prod_{i \in A_0} c_{v_i}^{\lambda_i}\right)^{-\gamma}$ .
3:   if  $\forall i \in A_0, p_i \geq C$  then
4:     Compute  $\Delta'_{A_0} = R_s - R_v \bmod P_{A_0}$ .
5:     Check  $\gamma \stackrel{?}{=} H_{\text{FS}}(c_s, \{c_{v_i}\}_{i \in A_0}, \bar{c}'_s, \bar{c}'_v, \Delta'_{A_0}, \text{com}_{\text{MT}})$ .
6:   else
7:     Compute  $\bar{c}'_k = g^{R_k} h^{R_{rk}} \cdot c_k^{-\gamma}$  and  $\Delta'_A = R_s - R_k P_{A_0} - R_v$ .
8:     Check  $\gamma \stackrel{?}{=} H_{\text{FS}}(c_s, \{c_{v_i}\}_{i \in A_0}, c_k, \bar{c}'_s, \bar{c}'_k, \bar{c}'_v, \Delta'_{A_0}, \text{com}_{\text{MT}})$ .
9:   end if
10:  Check the validity of the range proof  $1 \stackrel{?}{=} \text{RP.Verify}((c_s, B), \pi_{\text{RP}})$ .
11:  If any check fails, broadcasts complain about the dealer.
12: end for
13: for  $i \in A_0$  do
14:  Check the commitment consistency:  $1 \stackrel{?}{=} \text{IC.Verify}(c_{v_i}, (v_i, (v_i, 1, r_{v_i})))$ .
15:  If found any inconsistencies, broadcast complain about the dealer.
16: end for
17: for  $j \in A'_0$  do
18:  if  $p_j \geq C$  then
19:    Recompute  $\Delta'_j = R_s - R_v - \gamma v_j \lambda_j \bmod p_j$ .
20:    Define  $\text{tr}_j = (\Delta'_j, v_j)$  and check  $1 \leftarrow \text{MT.Verify}(\text{com}_{\text{MT}}, \text{tr}_j, \text{auth}_j, j)$ .
21:  else
22:    Recompute  $\bar{c}_{k_j} = g^{R_{k_j}} h^{R_{rk_j}} c_{k_j}^{-\gamma}$  and  $\Delta'_j = R_s - R_v - \gamma \lambda_j v_j - p_j R_{k_j}$ .
23:    Define  $\text{tr}_j = (c_{k_j}, \bar{c}_{k_j}, \Delta'_j, v_j)$  and check  $1 = \text{MT.Verify}(\text{com}_{\text{MT}}, \text{tr}_j, \text{auth}_j, j)$ .
24:  end if
25:  If any check fails, broadcast complain about the dealer.
26: end for

```

correctness of the CRT, the reconstruction returns a unique $s \in [0, B)$, and reducing modulo p_0 leads to the unique secret s_0 .

Secrecy. For any unauthorised set $\bar{A} \in \mathfrak{A}$, the total weight of parties in $\bar{A}$ is less than or equal to t (the privacy threshold). Let $s_0, s_1 \in \mathbb{F}_{p_0}$ be the challenge secrets. For simpler notation, we write $V_{\bar{A}}(s)$ for the corrupted CRT shares and $\text{View}_{\bar{A}}(s)$ for the public transcript and all private records of $\bar{A}$. Consider the following hybrids

- G_0 : The view of $\bar{A}$ based on sharing s_0 , or simply, $\text{View}_{\bar{A}}(s_0)$.
- G_1 : In this game, we keep the corrupted parties' shares $V_{\bar{A}}(s_0)$ unchanged and replace the verification information with simulated information.
 - **Simulate the Range Proof.** Replace the range proof π_{RP} for (c_s, B) with the output of its zero-knowledge simulator by applying the statistical response-simulation hybrids while keeping the witness-dependent commitments. By the SHVZK of the range proof, this replacement is statistically indistinguishable, with negligible error ϵ_{RP} .
 - **Simulate the UPoM proofs.** Invoke the simulators of UPoM and boosted UPoM jointly by applying their statistical response-simulation

hybrids while keeping all witness-dependent commitments fixed. Each shared response value is sampled once. All visible relation values, including the enforcement Δ , are then reconstructed deterministically from these simulated responses and the corrupted parties' shares. This replacement is statistically indistinguishable with negligible error ϵ_{UPoM} by Lemma 2 and Claim 1.

- **Hide the Unopened Merkle Leaves.** For every honest party $j \notin A'_0 \setminus \bar{A}$, replace the committed entry with a fixed dummy entry, while leaves, root, authentication paths disclosed to parties in $\bar{A} \cap A'_0$ remain generated from one consistent tree. Hence, generating these Merkle tree data no longer requires the honest parties' shares. By the hiding of the Merkle-tree-based vector commitment, this replacement is computationally indistinguishable with negligible error ϵ_{MT} .
- **Hide the Witness Commitments.** Keep the disclosed openings belonging to corrupted parties unchanged, while replacing the remaining witness commitment (including c_s , honest parties' commitments, and the range proof witness commitments) with commitments to zero. By the hiding property of the integer commitment scheme, this replacement is either statistical (in the trusted setup) or computational (in the transparent setup) with negligible error ϵ_{IC} .

Therefore, every part of the view can be generated from pp and $V_{\bar{A}}(s_0)$ solely, without depending on s_0 or honest parties shares. We have $G_0 \approx_c G_1$ except for simulation error $\epsilon_{\text{RP}} + \epsilon_{\text{UPoM}} + \epsilon_{\text{MT}} + \epsilon_{\text{IC}}$, which is negligible.

- G_2 : In this game, we run the same simulator in G_1 , but change the corrupted share vector from $V_{\bar{A}}(s_0)$ to $V_{\bar{A}}(s_1)$. By statistical privacy of the CRT secret sharing, we have $\text{SD}(V_{\bar{A}}(s_0), V_{\bar{A}}(s_1)) \leq \frac{2P_{\bar{A}}}{L} = \text{negl}(\lambda)$ (see [14, Claim 1]). Hence, $G_1 \approx_s G_2$ except with negligible statistical error.
- G_3 : In this game, we replace the simulated verification information in G_2 with honestly generated verification materials for sharing s_1 . Thus, $G_3 := \text{View}_{\bar{A}}(s_1)$. We keep the corrupted share vector $V_{\bar{A}}(s_1)$ unchanged and apply the simulation argument used in transition $G_0 \rightarrow G_1$ to the secret s_1 , such that $G_2 \approx_c G_3$ except for the distinguishing error of $\epsilon_{\text{RP}} + \epsilon_{\text{UPoM}} + \epsilon_{\text{MT}} + \epsilon_{\text{IC}} = \text{negl}(\lambda)$.

Commitment. Let $A, \hat{A} \in \mathfrak{A}$ be two authorised sets. Suppose the PPT adversary outputs a common broadcast transcript π and private records for A and $\hat{A}$ such that all records used by both reconstructions are accepted. We first argue that, except with negligible probability, there exists one integer $s^* \in [0, B)$ satisfying $s^* \equiv v_i \pmod{p_i}$ for every accepted $i \in A \cup \hat{A}$.

By the random oracle forking lemma and the special soundness of Π_{UPoM} and Π_{UPoMS} (Theorems 1 and 3), two accepting transcripts with the same first message and distinct challenges γ, γ' yield integer openings s^*, v^* , and in the soundness boost setting, k^* , of the fixed commitments (c_s, c_v, c_k) . Writing $\gamma^* = \gamma - \gamma'$, the special soundness gives $s^* \equiv v^* \equiv \sum_{i \in A_0} \lambda_i v_i \pmod{P_{A_0}}$. Together with binding of the integer commitment scheme and public CRT coefficients, for every accepted $i \in A_0$, we have $s^* \equiv v_i \pmod{p_i}$.

For each accepting $j \in A'_0$, by the binding property of the Merkle-tree-based vector commitment fixes the entry $(\Delta_j, v_j,)$ or $(c_{k_j}, \bar{c}_{k_j}, \Delta_j, v_j)$ under the broadcast root. This follows from collision-resistance of H_{leaf} and H_{MT} . If $p_j \geq C$, subtracting the two accepted verification equations gives $\gamma^*(s^* - v^* - \lambda_j v_j) \equiv 0 \pmod{p_j}$. Since $j \notin A_0$, the global CRT coefficients leads to $v^* \equiv 0 \pmod{p_j}$. Moreover, since $0 \leq |\gamma^*| \leq C \leq p_j$, γ^* is invertible mod p_j . Hence, we have $s^* \equiv v_j \pmod{p_j}$. Similar can be argued for $p_j < C$, where the extracted integer k_j^* from c_{k_j} leads to $s^* - v^* - \lambda_j v_j = p_j k_j^* \implies s^* \equiv v_j \pmod{p_j}$.

Finally, special soundness of the range proof extracts an opening $\tilde{s} \in [0, B)$ of the same commitment c_s . Binding of the integer commitment guarantees $\tilde{s} = s^*$, leading to $s^* \in [0, B)$. It follows that $s^* \equiv v_i \pmod{p_i}$ for all $i \in A$ and $s^* \equiv \hat{v}_i \pmod{p_i}$ for all $i \in \hat{A}$. Since $s^* < B \leq P_A, P_{\hat{A}}$, CRT uniqueness forces the reconstruction for each set to equal s^* .

Let $z_A := \text{Recon}(\text{pp}, \{i, v_i, m_i\}_{i \in A}, \pi)$. Consider the event

$$\text{Bad}_{\text{com}} := \{z_A \neq \perp \wedge z_{\hat{A}} \neq \perp \wedge z_A \neq z_{\hat{A}}\}.$$

Prior argument shows that Bad_{com} can occur only if at least one of the following event occurs:

- The Fiat-Shamir forking fails with probability at most ϵ_{FS} .
- UPoM or boosted UPoM special soundness fails with probability at most $\epsilon_{\text{UPoM}} + \epsilon_{\text{UPoMS}}$.
- Integer commitment binding fails with probability at most ϵ_{IC} .
- Position binding of the Merkle tree commitment fails with probability at most ϵ_{MT} .
- Range proof special soundness fails with probability at most ϵ_{RP} .

By a union bound, we have

$$\Pr[\text{Bad}_{\text{com}}] \leq \epsilon_{\text{FS}} + \epsilon_{\text{UPoM}} + \epsilon_{\text{UPoMS}} + \epsilon_{\text{IC}} + \epsilon_{\text{MT}} + \epsilon_{\text{RP}} = \text{negl}(\lambda).$$

□

6 Application: Distributed Key Generation

We show that our CRT-based VWSS naturally extends to distributed key generation (DKG) for DSA and ECDSA keys, which is one of the core applications of verifiable secret sharing. At first sight, a structural mismatch seems unavoidable: our construction is instantiated with groups of unknown order, whereas (EC)DSA keys are generated in discrete-logarithm groups of known prime order. Nevertheless, since the underlying unbounded proof-of-mod is inherently a proof of knowledge of exponent, we can incorporate an efficient exponent-equality verification into the protocol. Crucially, this augmentation preserves the practical efficiency of the proposed VWSS as it only adds one discrete logarithm group element to the broadcast message. In particular, we revise the VWSS algorithm for DKG with g_{DL} denotes the discrete-logarithm group generator and Commit denotes the discrete-logarithm commitment (e.g., Pedersen commitment [21]) as follows:

Algorithm 5 Distributed Key Generation

- 1: **for** $i \in [n]$ **do**
 - 2: Select $s_{i,0} \xleftarrow{\$} \mathbb{Z}_q$ and broadcast $c_i = H_{\text{dkg}}(g_{\text{DL}}^{s_{i,0}}, \rho_i)$ for $\rho_i \xleftarrow{\$} \{0, 1\}^{2\lambda}$.
 - 3: On receiving $\{c_j\}_{j \in [n] \setminus \{i\}}$ from other parties, broadcast the opening of $y_i = g_{\text{DL}}^{s_{i,0}}$.
 - 4: Lift $s_{i,0}$ to s_i according to the CRT-based secret sharing and perform the revised VWSS of s_i .
 - 5: Set pk to $y = \prod_{i=1}^n y_i$ and the corresponding secret key is $x = \sum_{i=1}^n s_{i,0} \bmod q$.
 - 6: Locally compute CRT-based secret share of x as $x_i = \sum_{i=1}^n v_i \bmod p_i$.
 - 7: **end for**
-

- Revise the CRT-based VWSS at Algorithm 3 as follows:
 - Compute $y = g_{\text{DL}}^{\bar{s}}$ at line 2.
 - Compute $\bar{y} = g_{\text{DL}}^{\bar{s}}$ at line 9, where $\bar{s}$ comes from $\bar{d}_s$.
 - Add $y, \bar{y}$ to the hash input in line 24 and line 28.
 - Add y to the broadcast message in line 26 and line 30.
- Revise the CRT-based VWSS verification at Algorithm 4 as follows:
 - Add $\bar{y}' = g_{\text{DL}}^{R_s} y^{-\gamma}$ at line 2.
 - Add y and $\bar{y}'$ to the hash input at line 5 and 8.

It is important to note that the secret key reconstruction suffers from an integer growth issue. Concretely, let x, y be lifted secrets shared under CRT-based VWSS. Their local addition may produce an integer $z = x + y$ that exceeds the reconstruction threshold. Let A be an authorised set of parties, when $z > P_A$, where $P_A = \prod_{i \in A} p_i$, correct reconstruction is no longer guaranteed. Nevertheless, local additions only incurs a polynomial growth on the size of lifted integers. Hence, as long as $T \geq \text{poly}(\lambda) \cdot 2^{t+2\lambda}$, where T denotes the reconstruction threshold and t is the privacy threshold, the integer growth in the DKG does not compromise the correctness. Accordingly, we set the reconstruction threshold to $T = 2^{t+3\lambda}$ in the DKG which handles share addition without violating correctness.

Theorem 6. *Assume that the revised CRT-based VWSS (Algorithms 3 and 4 with the modifications described above) satisfies the correctness, secrecy, and commitment properties of Theorem 5; H_{dkg} is modelled as a random oracle; and the discrete logarithm assumption holds in $\mathbb{G}_{\text{DL}}$. By setting $p_0 = q$ and $T = 2^{t+3\lambda}$, Algorithm 5 achieves correctness, secrecy, and commitment.*

Proof. For each $i \in [n]$, we refer to the i -th party as P_i . We write v_{ij} for the CRT share produced by the i -th VWSS instance for receiver j , corresponding to v_i in Algorithm 5 line 6. We write x_i for the accumulated CRT share of party i , corresponding to line 6 of Algorithm 5. Let $s_{i,0} \in \mathbb{Z}_{p_0}$ with $p_0 = q$ and $s_i = s_{i,0} + u_i p_0$ be the lifted CRT secret. Notably, each $s_{i,0}$ is a valid group element for the DKG.

Correctness. First, we show that $y = \prod_{i=1}^n y_i = g_{\text{DL}}^x$ where $x = \sum_{i=1}^n s_{i,0} \bmod q$. In line 2 (Algorithm 5), each party P_i broadcasts $c_i = \text{Commit}(g_{\text{DL}}^{s_{i,0}})$, then P_i

broadcasts the opening of $y_i = g_{\text{DL}}^{s_{i,0}}$ in line 3 (Algorithm 5). By the computational binding of **Commit**, $y_i = g_{\text{DL}}^{s_{i,0}}$ is uniquely defined at line 2. Therefore:

$$y = \prod_{i=1}^n y_i = g_{\text{DL}}^{\sum_{i=1}^n s_{i,0}} = g_{\text{DL}}^x$$

Then we prove that x_i is a valid CRT share of x , and x can be correctly recovered.

- 1 Each individual share v_{ij} is valid. In both cases, $s_j \in [0, B)$ is a valid VWSS secret. By VWSS correctness (Theorem 5), the revised Algorithm 3 correctly produces the CRT share $v_{ij} = s_j \bmod p_i$, where the lifting is: $s_j = s_{j,0} + u_j \cdot p_0$ ($u_j \xleftarrow{\$} [0, L)$) satisfies $s_j \bmod p_0 = s_{j,0}$.
- 2 Since $v_{ij} \equiv s_j \pmod{p_i}$ for each $j \in [n]$, by the linearity of CRT sharing:

$$x_i = \sum_{j=1}^n v_{ij} \bmod p_i$$

is a CRT-based secret share of $Z = \sum_{j=1}^n s_j = \sum_{j=1}^n (s_{j,0} + p_0 \sum_{j=1}^n (u_j) \equiv \sum_{j=1}^n s_{j,0} \bmod p_0$. Therefore, reconstructing Z allows parties to compute $x = \sum_{j=1}^n s_{j,0} \bmod p_0$.

- 3 $Z < P_A$ holds since

$$Z \leq n \cdot ((p_0 + 1)B) \leq \text{poly}(\lambda) \cdot 2^{t+2\lambda} \ll 2^{t+3\lambda} \leq P_A.$$

Hence, by applying the CRT, we have a unique solution of $Z = s = \sum_{j=1}^n s_j$ in $[0, P_A)$ when given $\{Z \equiv x_i \pmod{p_i}\}_{i \in A}$.

Finally, if at least one honest secret is uniformly distributed and no aborts in the subsequent protocol, then the generated secret key is uniformly distributed. **Secrecy.** Let $\bar{A}$ be an unauthorised set that is corrupted whose total weight is less than or equal to t (the privacy threshold). For simpler notation, we denote the corrupted shares of $s_{i,0}$ as $V_{i,\bar{A}}(s_{i,0})$ and the view of corrupted parties as $\text{View}_{i,\bar{A}}(s_{i,0})$. Consider the following hybrids:

- G_0 : The parties execute DKG protocol honestly, except for those controlled by the corrupted parties in $\bar{A}$.
- G_1 : For each honest dealer $P_i \notin \bar{A}$, replace its augmented VWSS transcript and auxiliary verification data by simulated values following the secrecy of the VWSS while leaving their shares for corrupted parties and the public contribution y_i untouched. We omit the detailed replacement as it directly follows the proof of Theorem 5. Note that $\bar{y}_i$ can be simulated with R_{s_i} . Under the corresponding simulation and hiding arguments, we have $G_0 \approx_c G_1$.
- G_2 : For every honest dealer $P_i \notin \bar{A}$, replace $V_{i,\bar{A}}(s_{i,0})$ by a fresh vector $V_{i,\bar{A}}(0)$ generated by sharing of 0. The public contribution remains y_i . Owing to the privacy of the CRT-based secret sharing, we have $\text{SD}(V_{i,\bar{A}}(s_{i,0}), V_{i,\bar{A}}(0)) \leq \frac{2P_{\bar{A}}}{L}$ [13, Claim 1]. Let $I = [n] \setminus \bar{A}$, we have $\text{SD}(G_1, G_2) = \frac{2|I|P_{\bar{A}}}{L} = \text{negl}(\lambda)$.

# Parties	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$
Total Weights	524	936	2490	4009	8834	18216
Reconstruction (T)	524	477	1269	2044	4505	9290
Core Proving Time (s)						
Method 1	0.225	0.263	0.451	0.675	1.296	2.534
Method 2	0.225	0.325	0.691	1.987	5.047	15.853
Verification Time (s)						
Method 1	0.232	0.254	0.601	1.192	3.612	11.829
Method 2	0.232	0.225	0.487	0.858	2.353	7.300
Broadcast Message (KB)						
Method 1	4.42	4.83	7.35	11.88	20.41	37.28
Method 2	4.42	3.70	5.54	7.66	13.55	25.47
Total Private Message (KB)						
Method 1	0.52	0.87	2.02	4.28	9.45	20.77
Method 2	0.52	3.33	7.01	23.75	55.70	164.63

Table 2. Performance of VWSS instantiated with different random weight vectors. The broadcast message includes all commitments.

- G_3 : For every honest party $i \in I$, sample an independent $c_i \xleftarrow{\$} \{0,1\}^{2\lambda}$ and broadcast c_i . After receiving all commitments, for each corrupted party $j \in \hat{A}$, search the random oracle query to obtain $y_j \in \mathbb{G}_{\text{DL}}$. Let $h \in I$, and for each $i \in I \setminus \{h\}$, choose $a_i \xleftarrow{\$} \mathbb{Z}_{p_0}$ and compute $y_i = g_{\text{DL}}^{a_i}$. Compute $y_h = y^* \left(\prod_{i \in I \setminus \{h\}} y_i \prod_{j \in \hat{A}} y_j \right)^{-1}$ such that $\prod_{l=1}^n y_l = y^*$. For each honest party i , sample a fresh ρ_i and program the random oracle such that $H_{\text{dkg}}(y_i, \rho_i) = c_i$ and broadcast (y_i, ρ_i) as the opening. Hence, we have $G_2 \approx_c G_3$ except for the programming error ϵ_{dkg} .

Commitment. Under the same assumption and adversary model as the VWSS commitment theorem, suppose an adversary outputs a common DKG transcript π and two collections of share records $Q_A, Q_{\hat{A}}$ for two authorised sets $A, \hat{A}$. Let $x := \text{Recon}_{\text{dkg}}(\text{pp}, Q_A, \pi)$ and $x' = \text{Recon}_{\text{dkg}}(\text{pp}, Q_{\hat{A}}, \pi)$. Assume both reconstruction succeed, we show that $x = x'$ except with negligible probability.

For each dealer $i \in [n]$, both collections are verified against the same VWSS transcript π_i contained in π . By the VWSS commitment property, there exists an integer $s_i^* \in [0, B)$ such that $v_{i,j} \equiv s_i^* \pmod{p_j}$ for $j \in A$ and $\hat{v}_{i,j} \equiv s_i^* \pmod{p_j}$ for $j \in \hat{A}$. Define $Z^* = \sum_{i=1}^n s_i^*$, then $x_j \equiv \sum_{i=1}^n v_{ij} \equiv Z^* \pmod{p_j}$ for $j \in A$ and $\hat{x}_j \equiv \sum_{i=1}^n \hat{v}_{ij} \equiv Z^* \pmod{p_j}$ for $j \in \hat{A}$. Moreover, we have

$$0 \leq Z^* \leq nB < P_{\min} \leq \min\{P_A, P_{\hat{A}}\}.$$

According to the uniqueness of the CRT and reducing modulo p_0 , we have $x = x' = Z^* \pmod{p_0}$, except for the error $n \cdot \Pr[\text{Bad}_{\text{com}}] = \text{negl}(\lambda)$.

□

7 Implementation and Benchmark

We implement the verifiable weighted secret sharing (VWSS) scheme in C++.⁶ Benchmarks were conducted on a Linux system equipped with an Intel i7-12700KF processor and 32 GB of memory. During benchmarking, Hyper-Threading and Turbo Boost were disabled, and the hash functions were not AVX-optimised. Furthermore, we instantiated the group of unknown order using a class group of an imaginary quadratic order with a 1600 bits discriminant, corresponding to 120 bits of security, following [4].

We benchmark the scheme using randomly generated weight vectors, as shown in Table 2. We set individual weights range from 10 to 256, and the number of parties varies from 4 to 128. Except for the $n = 4$ setting, the reconstruction threshold is set to 51% of the total weight. We evaluate the core proving time (UPoM and range proof generation time), verification time, broadcast communication, and total private communication.

We implement the three square decomposition required by the range proof [6] using a randomised probing algorithm. Since its running time varies significantly, we measure this step separately and exclude it from the timings reported in Table 2. For smaller parameter sets, its cost remains modest, where the average decomposition time is 0.022 seconds for $n = 4$ and 1.818 seconds for $n = 32$ within 100 independent runs. The cost becomes non-negligible as the parameters grow, reaching a average of 14.793 seconds for $n = 64$ and a average of 173.607 seconds for $n = 128$ within 10 independent runs.

We note that a direct comparison with the verifiable weighted secret sharing scheme of Shehata et al. [24] is not feasible, as the authors did not provide an open-source implementation, and their construction does not yield a concrete dealer runtime due to the prohibitive proof generation overhead for a large weight vector.

Remark 2. The randomised three square decomposition required by the range proof of Couteau et al. [6] can dominate the prover’s running time for large total weights. However, we note that the heaviest part of this computation can be performed offline, before the sharing step. Recall that their range proof for $s \in [0, B)$ requires computing integers (a_1, a_2, a_3) satisfying

$$a_1^2 + a_2^2 + a_3^2 = 4s(B - s) + 1.$$

Instead of proving the range of $s = s_0 + up_0$ directly, the prover separately commits to s_0 and u and proves that $s_0 \in [0, p_0)$ and $u \in [0, L)$. Since u is sampled independently of the secret s_0 , the prover can generate commitment to u , together with its range proof and three square decomposition, during offline preprocessing phase. During the online phase, the prover commits to s_0 and generate its three square decomposition and range proof, which is inexpensive. By the homomorphic property of the commitment scheme, the two commitments can be combined to obtain the commitment of s as

$$c_s = c_{s_0} \cdot c_u^{p_0}.$$

⁶ Implementation is available at <https://github.com/cryptome-xyz/CRT-VWSS>.

8 Conclusion

In this work, we develop new unbounded proof-of-mod protocols that serve as a foundation for constructing efficient verifiable weighted secret sharing schemes. Specifically, given the commitments to arbitrary integers s and v , our protocol enables a prover to efficiently show that $s \equiv v \pmod{N}$. While the basic construction has soundness that depends on the size of the smallest factor of N , we show how to reduce the soundness error by having the prover additionally commit to an integer $k \in \mathbb{Z}$ where $s = v + kN$. Building on unbounded proof-of-mod protocols, we construct the first practical verifiable weighted secret sharing (VWSS) scheme which requires the prover to broadcast only $O(|A|)$ commitments together with proof messages for a subset of parties A whose total weights meets the reconstruction threshold. Our approach leverages a novel structural property of CRT-based secret sharing, namely, the existence of global CRT coefficients. To this end, we eliminate the need for the dealer to generate separate proof-of-mod instances per participant. This significantly reduces the computational overhead of the dealer. For a weighted system with 4 parties and total weights 524, our VWSS allows the dealer to generate the proof with 0.225 seconds, which is orders of magnitude faster than the prior CRT-based verifiable weighted secret sharing [24]. Finally, we design a distributed key generation protocol for (EC)DSA keys using the VWSS as a building block, demonstrating its applicability in real-world settings.

Acknowledgments.

This project is supported by Cobo Singapore, and the Australian Research Council (ARC) Discovery Project DP260100245.

References

1. Boneh, D., Bünz, B., Fisch, B.: Batching techniques for accumulators with applications to iops and stateless blockchains. In: *Crypto*. pp. 561–586. Springer (2019)
2. Breidenbach, L., Cachin, C., Chan, B., Coventry, A., Ellis, S., Juels, A., Koushanfar, F., Miller, A., Magauran, B., Moroz, D., et al.: Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chainlink Labs* **1**(1-136), 2 (2021)
3. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: *S&P*. pp. 315–334. IEEE (2018)
4. Bünz, B., Fisch, B., Szepieniec, A.: Transparent snarks from dark compilers. In: *Eurocrypt*. pp. 677–706. Springer (2020)
5. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: *SFCS*. pp. 383–395. IEEE (1985)
6. Couteau, G., Klooß, M., Lin, H., Reichle, M.: Efficient range proofs with transparent setup from bounded integer commitments. In: *Eurocrypt*. pp. 247–277. Springer (2021)

7. Damgård, I., Fujisaki, E.: A statistically-hiding integer commitment scheme based on groups with hidden order. In: *Asiacrypt*. pp. 125–142. Springer (2002)
8. Desmedt, Y.: Society and group oriented cryptography: A new concept. In: *conference on the theory and application of cryptographic techniques*. pp. 120–127. Springer (1987)
9. Desmedt, Y.G., Frankel, Y.: Threshold cryptosystems. In: *Proceedings on Advances in cryptology*. pp. 307–315 (1989)
10. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*. pp. 427–438. IEEE (1987)
11. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: *Eurocrypt*. pp. 186–194. Springer (1986)
12. Fu, S.: Inner-product commitments over integers with applications to succinct arguments. In: *Asiacrypt*. pp. 521–553. Springer (2025)
13. Garg, S., Jain, A., Mukherjee, P., Sinha, R., Wang, M., Zhang, Y.: Cryptography with weights: Mpc, encryption and signatures. In: *Crypto*. pp. 295–327. Springer (2023)
14. Garg, S., Kolonelos, D., Policharla, G.V., Wang, M.: Threshold encryption with silent setup. In: *Crypto*. pp. 352–386. Springer (2024)
15. Gennaro, R., Goldfeder, S.: Fast multiparty threshold ecDSA with fast trustless setup. In: *ACM CCS*. pp. 1179–1194 (2018)
16. Grandjean, D., Heimbach, L., Wattenhofer, R.: Ethereum proof-of-stake consensus layer: Participation and decentralization. In: *FC*. pp. 253–280. Springer (2024)
17. Kaya, K., Selçuk, A.A.: Secret sharing extensions based on the chinese remainder theorem. *Indocrypt* (2010)
18. Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: *Asiacrypt*. pp. 598–616. Springer (2009)
19. Merkle, R.C.: A digital signature based on a conventional encryption function. In: *Eurocrypt*. pp. 369–378. Springer (1987)
20. Micali, S., Goldreich, O., Wigderson, A.: How to play any mental game. In: *STOC*. pp. 218–229. ACM New York (1987)
21. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: *Crypto*. pp. 129–140. Springer (1991)
22. Schindler, P., Judmayer, A., Stifter, N., Weippl, E.: Hydrand: Efficient continuous distributed randomness. In: *S&P*. pp. 73–89. IEEE (2020)
23. Shamir, A.: How to share a secret. *Communications of the ACM* **22**(11), 612–613 (1979)
24. Shehata, K., Fangqi, H., Thyagarajan, S.A.: Verifiable weighted secret sharing. *Crypto Valley Conference* (2025)
25. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. pp. 256–266. Springer (1997)
26. Yao, A.C.C.: How to generate and exchange secrets. In: *27th annual symposium on foundations of computer science (Sfcs 1986)*. pp. 162–167. IEEE (1986)