BulletCT: Towards More Scalable Ring Confidential Transactions With Transparent Setup

Nan Wang[1,*], Qianhui Wang[2], Dongxi Liu[1], Muhammed F. Esgin[3], Alsharif Abuadbba[1]
CSIRO's Data61[1], *University of Cambridge*[2], *Monash University*[3]

Abstract

RingCT signatures are essential components of Ring Confidential Transaction (RingCT) schemes on blockchain platforms, enabling anonymous transaction spending and significantly impacting the scalability of these schemes. This paper makes two primary contributions:

We provide the first thorough analysis of a recently developed *Any-out-of-N proof* in the discrete logarithm (DLOG) setting and the associated RingCT scheme, introduced by ZGSX23 (S&P '23). The proof conceals the number of the secrets to offer greater anonymity than *K-out-of-N proofs* and uses an efficient "K-Weight" technique for its construction. However, we identify for the first time several limitations of using Any-out-of-N proofs, such as increased transaction sizes, heightened cryptographic complexities and potential security risks. These limitations prevent them from effectively mitigating the longstanding scalability bottleneck.

We then continue to explore the potential of using K-out-of-N proofs to enhance scalability of RingCT schemes. Our primary innovation is a new DLOG-based RingCT signature that integrates a refined "K-Weight"-based K-out-of-N proof and an entirely new tag proof. The latter is the first to efficiently enable the linkability of RingCT signatures derived from the former, effectively resisting double-spending attacks.

Finally, we identify and patch a linkability flaw in ZGSX23's signature. We benchmark our scheme against this patched one to show that our scheme achieves a boost in scalability, marking a promising step forward.

1 Introduction

In the realm of cryptography, zero-knowledge proofs have emerged as a powerful tool for verifying the authenticity of public statements in a privacy-preserving manner. They allow a prover to convince a verifier that a statement is true without revealing other private information. The development of partial knowledge proofs represents a significant advancement, which allows a party knowing witnesses for some subset of public statements to convince others of the claim without revealing which subset. They are crucial building blocks of Ring Confidential Transaction (RingCT) schemes on blockchain systems. RingCT schemes have emerged as a prominent solution, enabling decentralized transaction spending in an anonymous and confidential manner.

Bottleneck. Poor scalability has been a long-standing major bottleneck for all blockchain systems, where the scalability refers to the ability to handle an increasing number of transactions. Presently, blockchain platforms are still significantly less scalable than centralized platforms due to their decentralized nature. Tackling scalability challenges is a long-term goal for unleashing the full potential of blockchains, improving their sustained viability and widespread adoption across various sectors. However, scalability challenges are significantly exacerbated in RingCT schemes due to the involved computational and communication-intensive zero-knowledge proofs for ensuring valid transaction spending.

Critical Metrics. Transactions Per Second (TPS) and Time-To-Finality (TTF) are two essential metrics for evaluating blockchain performance [1, 19]. TPS measures the number of transactions processed within a given timeframe, while TTF refers to the time required to finalize a transaction with no risk of reversal. Higher TPS and shorter TTF indicate greater scalability. In RingCT schemes, both TPS and TTF are significantly affected by transaction sizes and the verifier costs associated with zero-knowledge proofs. Smaller transactions can increase the number of transactions per block and contribute to lower transaction fees, as they have a competitive advantage when vying to be part of the next block on the blockchain [16]. Meanwhile, higher verifier efficiency help boost the number of transactions processed within a given timeframe and considerably reduces transaction finalization time. Thus, reducing transaction sizes and increasing verifier efficiency are two effective ways to enhance scalability.

Research Gap. The state-of-the-art RingCT schemes, including Monero [14], Omniring (CCS '19) [15], RingCT-3.0 (FC

*The corresponding author.

'20) [24], and ZGSX23 (S&P '23) [25] have faced scalability challenges since their inception, primarily due to their inherent performance trade-offs resulting in either larger transaction sizes or higher verification costs. Additionally, trusted setups are problematic because they require a group of parties to jointly generate public parameters using trapdoor information, which undermines the decentralization of blockchain systems. Although transparent setups address the trapdoor issue, they complicate the design of efficient RingCT schemes. Thus, this leaves us an intriguing and challenging question:

Question

Is there a way to construct a more scalable RingCT scheme enjoying both smaller transactions and higher verifier efficiency without using a trusted setup?

2 Background

To lay the groundwork for our contributions, we begin with a brief overview of the typical RingCT scheme, Omniring, to provide foundational insights into general RingCT schemes.

RingCT Scheme. A RingCT scheme is a cryptographic protocol used primarily in privacy-focused cryptocurrencies. It consists of a transaction model and a tuple of probabilistic polynomial-time (PPT) algorithms, including *Setup*, *KeyGen*, *AccountGen*, *Spend* and *Receive*. In this work, we will focus on the *spend* phase.

Transaction Model. A transaction $\mathsf{tx} := (\mathcal{R}, \mathcal{S}, \mathcal{T}, \sigma)$ is composed of:

- a set of ring accounts $\mathcal{R} = (\mathsf{acc}_i)_{i=1}^{|\mathcal{R}|}$
- a set of source accounts' tags $\mathcal{S} = (T_k)_{k=1}^{|\mathcal{S}|}$
- a set of target accounts $\mathcal{T} = (\mathsf{acc}_j)_{j=1}^{|\mathcal{T}|}$
- a RingCT signature σ

In each transaction, a user mixes up $|\mathcal{S}|$ number of source accounts with $(|\mathcal{R}| - |\mathcal{S}|)$ number of *decoy* accounts to create an ad-hoc set of accounts known as a *ring*, where $|\cdot|$ is the cardinality of a set. $|\mathcal{R}|$ is typically far larger than $|\mathcal{S}|$ to offer strong anonymity. The user spends the hidden source accounts and creates new target accounts, where each account $\mathsf{acc} := (P, A)$ is comprised of a public key $P = \tau^s$ concealing a non-zero secret key s and an amount commitment A hiding a non-negative value to be transferred. Meanwhile, the user must reveal a unique tag T derived from one of the source account's public key. Validators on blockchain maintain a set of previously revealed tags as "state" and any transaction revealing a tag already in this set is flagged as a double-spending attack. It is one of the most critical threats to RingCT schemes as it allows a user to spend one account multiple times, effectively creating new values illegally. Furthermore, each transaction only includes the offset values of the account ring $|\mathcal{R}|$, which reference the target accounts from historical transactions for storage efficiency.

RingCT Signature. A RingCT signature σ is an essential component empowering a user to legally sign and spend a transaction, also known as a signed "spend proof". Omniring defines a language $\mathcal{L}_K$ outlining the constraints a RingCT signature must meet when using a K-out-of-N proof:

$$\mathcal{L}_K \triangleq \begin{cases} \mathsf{stmt} = \left((P_i, A_i^{\mathcal{R}})_{i=1}^{|\mathcal{R}|}, (T_k)_{k=1}^{|\mathcal{S}|}, (A_j^{\mathcal{T}})_{j=1}^{|\mathcal{T}|}\right) : \\ \exists \mathsf{wit} = \left((s_{\phi(k)}, a_{\phi(k)}, r_{\phi(k)})_{k=1}^{|\mathcal{S}|}, (t_j, m_j)_{j=1}^{|\mathcal{T}|}\right) \text{ s.t. :} \\ \forall\, k \in \{1, \ldots, |\mathcal{S}|\} : \begin{cases} P_{\phi(k)} = \tau^{s_{\phi(k)}},\ T_k = \eta^{s^{-1}_{\phi(k)}}, \\ A^{\mathcal{R}}_{\phi(k)} = g^{a_{\phi(k)}} \rho^{r_{\phi(k)}} \end{cases} \\ \forall\, j \in \{1, \ldots, |\mathcal{T}|\} : \left(A_j^{\mathcal{T}} = g^{t_j} \rho^{m_j},\ t_j \in [0, 2^\beta - 1]\right) \\ \sum_{k=1}^{|\mathcal{S}|} a_{\phi(k)} = \sum_{j=1}^{|\mathcal{T}|} t_j \end{cases}$$

where g, τ, η, ρ are randomly sampled generators from a cyclic group $\mathbb{G}$ of prime order.

A RingCT signature is a combination of multiple zero-knowledge proofs to jointly fulfill the constraints in $\mathcal{L}_K$, including:

- A **K-out-of-N proof**, to prove the knowledge of the source public keys $(P_{\phi(k)})_{k=1}^{|\mathcal{S}|}$ out of a public-key ring $(P_i)_{i=1}^{|\mathcal{R}|}$. Let $\phi(k) \to i$ be an injective function mapping a space $k \in (1, \ldots, |\mathcal{S}|)$ to another space $i \in (1, \ldots, |\mathcal{R}|)$. Throughout the paper, we assume $|\mathcal{R}| = N$ and $|\mathcal{S}| = K$.
- A **tag proof**, to prove that the tags $(T_k)_{k=1}^{|\mathcal{S}|}$ form a bijection to the source public keys $(P_{\phi(k)})_{k=1}^{|\mathcal{S}|}$. This proof is vital for enabling linkability of RingCT signatures. In Omniring, the bijection is strictly *order-preserving*, meaning that the k-th tag T_k exactly corresponds to the k-th source public key $P_{\phi(k)}$. However, a more flexible bijection in arbitrary order is already adequate to resist double-spending attacks and we will use this bijection in our tag proof.
- A **balance proof**, to prove that the sum of the source accounts' amounts is equal to that of the target accounts' amounts $\sum_{k=1}^{|\mathcal{S}|} a_{\phi(k)} = \sum_{j=1}^{|\mathcal{T}|} t_j$.
- $|\mathcal{T}|$ **number of range proofs**, to prove that the amounts of the target accounts $(t_j)_{j=1}^{|\mathcal{T}|}$ are all non-negative.

The mixture of these *complete*, *sound* and *zero-knowledge* proofs guarantee the three key security properties of RingCT spending, as defined and proven by Omniring [15]:

- **Balance:** It ensures that no users can spend more than they possess.
- **Privacy:** It captures the anonymity of spenders and the confidentiality of amount values being transferred.

Table 1: Key Symbols and Notations

Symbols	Descriptions
$\mathcal{P}$ & $\mathcal{V}$	The prover and verifier
$\mathcal{R}$	The account ring
$\mathcal{S}$ & $\mathcal{T}$	The set of source and target accounts
$\lvert\cdot\rvert$	The cardinality of a set or a vector
P_i	The i-th public key
s_i	The i-th secret key
T_i	The tag of the i-th account
β	The bit length of range proofs
$\phi()$	The injective function

- **Non-slanderability:** It prevents malicious users from authorizing the spending on behalf of others.

Advancements. The leading RingCT schemes have prioritized compactness in their RingCT signatures, aiming to enhance scalability of underlying RingCT schemes through smaller transactions. Monero combines *Multilayer Linkable Spontaneous Anonymous Group* signature [20] with Bulletproofs to construct RingCT signatures, achieving a communication complexity of $O(|\mathcal{R}||\mathcal{S}|+\log(\beta|\mathcal{T}|)$. The three following schemes surpass Monero by leveraging Bulletproofs' techniques (S&P '18) [5] to develop various DLOG-based partial knowledge proofs, enabling seamless integration with Bulletproofs' range instances to construct compact RingCT signatures. For example, Omniring and RingCT-3.0 use different K-out-of-N proofs. In contrast, ZGSX23 introduced an Any-out-of-N proof for constructing RingCT signatures, enhancing the anonymity of RingCT schemes by concealing $|\mathcal{S}|$. Anonymity levels of two ringsets are quantified using "anonymity spaces" as outlined in Table 2 based on their study. Notably, the anonymity level of Any-out-of-N proofs surpasses those of K-out-of-N proofs. Specifically, for ringset-II, using an Any-out-of-N proof with a 64-element ring achieves a comparable level of anonymity to the K-out-of-N proofs with a 116- and a 256-element ring in Omniring and RingCT-3.0, respectively.

Table 2: The anonymity spaces of partial knowledge proofs. Monero's default ring size 16 is used for RingCT-3.0 since they both use multiple small rings to support multiple source accounts, where one input is allocated to a separate ring.

Type	Omnring [15]	RingCT-3.0 [24]	ZGSX23 [25]
Space	$\binom{N}{K}=\frac{N!}{K!(N-K)!}$	16^K	2^N
Ringset-I	$\binom{64}{8}$, $N=64$	16^8, $N=128$	2^{32}, $N=32$
Ringset-II	$\binom{116}{16}$, $N=116$	16^{16}, $N=256$	2^{64}, $N=64$

High-Level Intuition. The three DLOG-based partial knowledge proofs build upon Bulletproofs' bit-vector technique. This technique is generally used in constructing zero-knowledge range proofs, where the prover demonstrates the knowledge of a bit vector that represents the committed non-negative value. In partial knowledge proofs, the bit vector is used to selectively retain the source accounts while discarding the others. Specifically, RingCT-3.0 designed a 1-out-of-N proof by using an N-bit vector with the Hamming weight of 1. Multiple 1-out-of-N proofs can be efficiently aggregated to build a K-out-of-KN proof by utilizing Bulletproofs' compression technique, where each proof has a different ring. Omniring's proof employs K number of N-bit vectors, where the Hamming weight of each vector is 1, to filter a single ring. Its proof creates two secret key vectors $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$ and $(s_{\phi(k)}^{-1})_{k=1}^{|\mathcal{S}|}$ hidden in the source public keys and tags along with two "weighted sum" values $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$ and $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}^{-1}$ for an arbitrary y to demonstrate the knowledge of multiple secret keys. Omniring constructs multiple computationally-intensive constraints to establish relations between the secret key vectors and "weighted sum" values. Instead, ZGSX23 employs a "K-Weight" technique to build an Any-out-of-N proof, enabling the prover to use a single bit vector with a secret Hamming weight of K to generate two "weighted sum" values. This approach offers the advantage of using a smaller bit vector, thereby reducing the number of required witnesses and lowering both computational and communication costs compared to Omniring's approach.

3 Contributions

3.1 An Analysis of Any-out-of-N proofs

We provide the first thorough analysis of a recently developed Any-out-of-N proof and the associated RingCT scheme, introduced by ZGSX23 (S&P '23) [25]. While using an Any-out-of-N proof enhances the anonymity of RingCT schemes, we identified several limitations that had not been adequately considered previously. These limitations prevent it from addressing the longstanding scalability bottleneck:

- **Increased Transaction Size.** Unlike K-out-of-N proofs, using an Any-out-of-N proof requires to use $|\mathcal{R}|$ tags rather than $|\mathcal{S}|$ tags in each transaction as $|\mathcal{S}|$ must remain confidential. The communication cost of the additional $(|\mathcal{R}|-|\mathcal{S}|)$ tags far outweighs the savings from the compact signature leading to increased transaction sizes since $|\mathcal{R}|$ is typically much larger than $|\mathcal{S}|$ to ensure strong anonymity.

- **Heightened Cryptographic Complexities.** To reduce on-chain "state" storage costs associated with the tag proliferation, an accumulator is employed to build a stateless[1] tagging scheme. However, this introduces several cryptographic complexities, including the need for

[1] Stateless blockchains are designed to eliminate the need for nodes to store "state" information. In RingCT schemes, nodes maintain a set of tags as "states", representing spent accounts for double-spending checks.

trusted setups, assumption incompatibility, and the associated security risks. Particularly, the assumption incompatibility gives rise to increased overheads. Moreover, whether stateless blockchains are more beneficial than stateful ones remains a controversial issue [7].

Note that the first limitation is inherent in Any-out-of-N proofs, regardless of the techniques employed, and is highly likely unavoidable. While the second one has a lesser impact on scalability, it still exerts a negative influence on the overall efficiency and security of schemes. Thus, at this point, our analysis indicates that Any-out-of-N proofs are less desirable for DLOG-based RingCT schemes than K-out-of-N proofs since the limitations far outweigh their benefits.

3.2 New RingCT Signature

Hence, we continue to explore the potential of using K-out-of-N proofs to create a new compact DLOG-based RingCT signature without using a trusted setup. Our construction aims to enable a more scalable RingCT scheme, *BulletCT*, to effectively mitigate the long-standing scalability bottleneck. At a high level, our contributions include a refined "K-Weight"-based K-out-of-N proof and a novel tag proof. Our tag proof overcomes a technical hurdle in establishing the linkability for RingCT signatures constructed from the former.

3.2.1 "K-Weight"-based K-out-of-N Proof

First, we leverage this "K-Weight" technique to develop a K-out-of-N proof by proving the Hamming weight of the bit vector is equal to the public $|\mathcal{S}|$. We refine ZGSX23's Any-out-of-N proof, enabling more seamless integration with Bulletproofs for signature construction while also reducing the number of elements by one.

3.2.2 New Tag Proof

Challenge. The primary challenge for constructing RingCT signatures is enabling linkability to prevent double-spending attacks. Omniring and RingCT-3.0 use a tag instantiation whose secret key is the reciprocal[2] of that of the corresponding source public key on a different generator. Their tag proofs employ the constraint $s \cdot s^{-1} = 1$ to establish separate reciprocal relations for multiple secret keys. However, this constraint does not apply to "K-Weight"-based proofs because it is exceptionally challenging to establish reciprocal relations using only the two weighted sums, $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$ and $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}^{-1}$ without relying on the secret key vectors. Additionally, we discovered that ZGSX23 did not offer a feasible tag proof to address this challenge.

[2]For brevity, the term "reciprocal" refers to the modular multiplicative inverse throughout the paper.

Solution. To address this challenge, we adopt an alternative strategy instead of the reciprocal one: we allow the prover to show that the same "weighted sum" $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$ can be extracted from both the public-key ring and tags. The key advantage is twofold: first, the proof requires only a single "weighted sum"; second, it eliminates the need for additional secret key vectors and their associated constraints, greatly reducing the computational costs. To achieve this, we use a different tag instantiation that shares the same secret key as the corresponding source public key on a different generator. Our approach patches the linkability flaw in ZGSX23's signature where the public-key ring and tags are of an equal size $|\mathcal{R}|$. However, when using "K-Weight"-based K-out-of-N proofs, a critical challenge arises due to the mismatch between the public-key ring's size $|\mathcal{R}|$ and the tags' size $|\mathcal{S}|$, complicating the extraction of the same "weighted sum" from both sets. We will explore this issue and present our solution in Section 6, following the necessary preliminaries.

3.2.3 Instantiations

In line with advanced RingCT schemes, we leverage Bulletproofs' techniques to create two instantiations of our proofs that seamlessly integrate with Bulletproofs, enhancing the compactness of the resulting RingCT signature. Compared to the patched ZGSX23, BulletCT eliminates the need for trusted setups and achieves a notable improvement in communication efficiency for scalability enhancement. We experimentally benchmarked our scheme against the patched scheme and show that our signature can help create a more scalable RingCT scheme.

3.2.4 Efficiency Comparisons

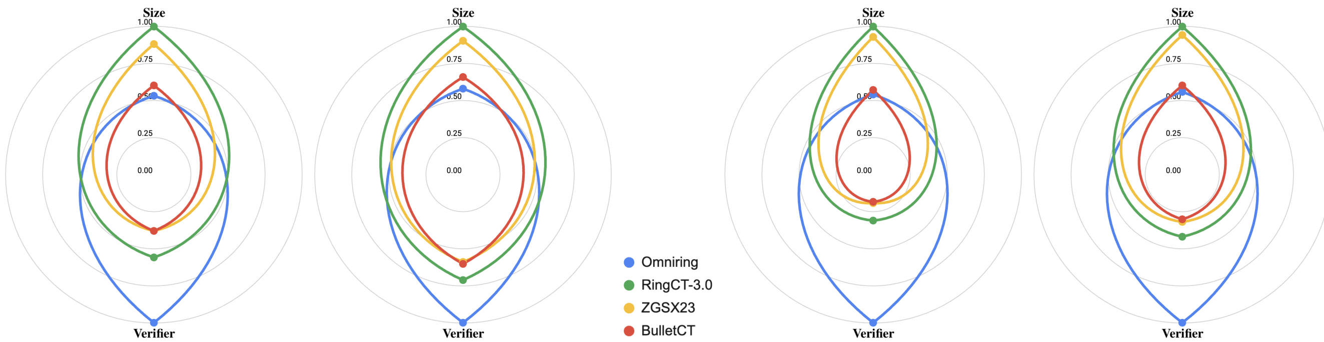
Throughout the paper, we use elliptic curve groups of a prime 256-bit order to instantiate zero-knowledge proofs, where the group and field elements have roughly the same size.

Complexity Comparisons. We essentially compared the involved group exponentiations as they dominate the computational costs. We provide a complexity comparison of state-of-the-art DLOG-based RingCT schemes in Table 3 in terms of the two critical metrics, transaction sizes and verifer costs. Notably, using an Any-out-of-N proof requires $|\mathcal{R}|$ tags in transactions.

Normalized Comparisons. Furthermore, to facilitate clearer comparisons, we present a normalized comparison of the two aforementioned ringsets using radar charts, as shown in Figure 1. BulletCT is consistently positioned at the innermost part of all four radar charts, demonstrating superior overall efficiency and establishing a solid foundation for scalability improvements. It can be observed that, Omniring, RingCT-3.0, and ZGSX23 all exhibit trade-offs in performance, resulting in either increased transaction sizes or reduced verifier efficiency. Specifically, ZGSX23 achieves high verifier efficiency

Table 3: An approximate complexity comparison of the state-of-the-art DLOG-based RingCT schemes, where β is the bit length for range proofs and $\delta = \beta|\mathcal{T}| + |\mathcal{R}||\mathcal{S}| + |\mathcal{R}| + 3|\mathcal{S}| + 3$. The costs associated with RSA groups are proportionally converted to those of DLOG groups. $\widetilde{\mathcal{R}}$ refers to pointers to ring accounts $\mathcal{R}$. We assume $|\mathcal{R}|$, β, $|\mathcal{S}|$ and $|\mathcal{T}|$ are the powers of 2 and can pad with zeros if not.

Type	Transaction Size (Elements)	Verifier Costs (Exps $\mathbb{G}$)
Omniring (K/N)	$\vert\mathcal{S}\vert+2\vert\mathcal{T}\vert+2\lceil\log\delta\rceil+9+\vert\widetilde{\mathcal{R}}\vert$	$(2\beta+1)\vert\mathcal{T}\vert+2\vert\mathcal{R}\vert\vert\mathcal{S}\vert+2\vert\mathcal{R}\vert+6\vert\mathcal{S}\vert+2\lceil\log\delta\rceil+6$
RingCT-3.0 (K/N)	$2\vert\mathcal{S}\vert+2\vert\mathcal{T}\vert+2\lceil\log(\vert\mathcal{R}\vert\beta\vert\mathcal{T}\vert\vert)\rceil+28+\vert\widetilde{\mathcal{R}}\vert$	$(2\beta+1)\vert\mathcal{T}\vert+4\vert\mathcal{R}\vert+\vert\mathcal{S}\vert+2\lceil\log(\vert\mathcal{R}\vert\beta\vert\mathcal{T}\vert\vert)\rceil+28$
ZGSX23 (Any/N)	$\vert\mathcal{R}\vert+2\vert\mathcal{T}\vert+2\lceil\log(\beta\vert\mathcal{T}\vert+\vert\mathcal{R}\vert)\rceil+11+(14\text{ RSA})+\vert\widetilde{\mathcal{R}}\vert$	$(2\beta+1)\vert\mathcal{T}\vert+6\vert\mathcal{R}\vert+2\lceil\log(\beta\vert\mathcal{T}\vert+\vert\mathcal{R}\vert)\rceil+(\vert\mathcal{R}\vert\text{ Hash})+12$
BulletCT (K/N)	$\vert\mathcal{S}\vert+2\vert\mathcal{T}\vert+2\lceil\log(\beta\vert\mathcal{T}\vert+\vert\mathcal{R}\vert+\vert\mathcal{S}\vert)\rceil+19+\vert\widetilde{\mathcal{R}}\vert$	$(2\beta+1)\vert\mathcal{T}\vert+4\vert\mathcal{R}\vert+3\vert\mathcal{S}\vert+2\lceil\log(\beta\vert\mathcal{T}\vert+\vert\mathcal{R}\vert+\vert\mathcal{S}\vert)\rceil+17$



(a) Ringset-I, $|\mathcal{T}| = 2$. (b) Ringset-I, $|\mathcal{T}| = 8$. (c) Ringset-II, $|\mathcal{T}| = 2$. (d) Ringset-II, $|\mathcal{T}| = 8$.

Figure 1: The normalized comparisons of various RingCT schemes by utilizing the two example ringsets in Table 2 to ensure a fair comparison at a comparable level of anonymity, where the anonymity spaces of our scheme is the same as those of Omniring. We use $\beta = 64$ and $|\widetilde{\mathcal{R}}| = \frac{|\mathcal{R}|}{8}$ since we allocate 4 bytes for each pointer. The closer the vertices to the center, the higher efficiency.

but suffers from substantial communication overhead due to its large tag sizes. Notably, we can observe that even with a smaller ring size $|\mathcal{R}| = 32$ in ringset-I which minimally impacts the overall transaction size, our transactions remain much smaller than theirs. Although Omniring excels in communication efficiency, it incurs the highest verifier costs due to the need for more computationally intensive constraints to establish connections between the secret key vectors and the corresponding weighted sums. RingCT-3.0 shows the largest transactions sizes with moderate verifier costs. In a nutshell, our scheme enjoys the efficiency advantages of both Omniring and ZGSX23. In Section 8, we will present a more comprehensive efficiency comparison of BulletCT and the patched ZGSX23, including prover time.

3.3 Roadmap

First, we introduce the cryptographic preliminaries in Section 4. We conduct a thorough review of Any-out-of-N proofs and highlight the limitations in Section 5. We elaborate on the technical challenges and specifics of our new tag proof in Section 6. Next we present the Bulletproofs-based instantiations of our K-out-of-N proof and tag proof in Section 7. We illustrate the benchmarks of our scheme against the patched ZGSX23 in Section 8.

4 Preliminaries

4.1 Notations

Let λ and $\text{negl}(\lambda)$ be the security parameter and a negligible function. Denote a cyclic group of prime order p by $\mathbb{G}$, and the ring of integers modulo p by $\mathbb{Z}_p$. Let $\mathbb{Z}_p^*$ be $\mathbb{Z}_p \setminus \{0\}$. Let $g, h, (g_i)_{i=1}^N, (h_i)_{i=1}^N \xleftarrow{\$} \mathbb{G}$ and $x \xleftarrow{\$} \mathbb{Z}_p^*$ be uniformly random generators and scalars from $\mathbb{G}$ and $\mathbb{Z}_p^*$, respectively. Denote the vector spaces of dimension N over $\mathbb{G}$ and $\mathbb{Z}_p$ by $\mathbb{G}^N$ and $\mathbb{Z}_p^N$, respectively. Bold font denotes vectors or matrices. For example, $\mathbf{a} = (a_1, \ldots, a_N) \in \mathbb{Z}_p^N$ and $\mathbf{g} = (g_1, \ldots, g_N) \in \mathbb{G}^N$ denote a vector of scalars and generators, respectively. $|\mathbf{a}|$ denotes the dimension of the vector $\mathbf{a}$. We define some basic vector operations below:

$$\mathbf{c} = \mathbf{a} + (\circ)\ \mathbf{b} = \big(a_1 + (\circ)\ b_1, \ldots, a_N + (\circ)\ b_N\big) \in \mathbb{Z}_p^N$$

$$g' = \boldsymbol{g}^{\mathbf{a}} = \prod_{i=1}^{N} g_i^{a_i} \in \mathbb{G}$$

$$\mathbf{g}' = \mathbf{g}_L \circ \mathbf{g}_R = (g_1 \cdot g_{\frac{N}{2}}, \ldots, g_{\frac{N}{2}+1} \cdot g_N) \in \mathbb{G}^N$$

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{N} a_i \cdot b_i \in \mathbb{Z}_p$$

$$\mathbf{a} \,||\, \mathbf{b} = (a_1,...,a_N,b_1,...,b_N) \in \mathbb{Z}_p^{2N}$$

where $\mathbf{g}_L = (g_1,...,g_{\frac{N}{2}})$ and $\mathbf{g}_R = (g_{\frac{N}{2}+1},...,g_N)$ are left-half and right-half sub-vectors. $\circ$, $\langle\cdot,\cdot\rangle$ and $||$ denote the Hadamard product, inner product and concatenation operations.

4.2 Assumption

Our trustless RingCT signatures rely on the hardness of discrete logarithm assumption:

Definition 1 (Discrete Logarithm (DLOG)) *The discrete logarithm assumption holds for all PPT adversaries $\mathcal{A}$:*

$$Pr\left[\begin{array}{l|l} (x_i)_{i=1}^N \leftarrow \mathcal{A}\big((g_i)_{i=1}^N\big), & \mathbb{G} \leftarrow \mathcal{G}(\lambda), \\ \prod_{i=1}^N g_i^{x_i} = \eta & (g_i)_{i=1}^N, \eta \xleftarrow{\$} \mathbb{G} \end{array}\right] \le \mathrm{negl}(\lambda)$$

where $\mathcal{G}(\lambda)$ is the setup algorithm. The assumption states that no computationally bounded adversaries can find such non-trivial discrete logarithm relations that satisfy $\prod_{i=1}^N g_i^{x_i} = \eta$ for an arbitrary $\eta \in \mathbb{G}$ and randomly chosen generators.

4.3 Homomorphic Commitment Schemes

Homomorphic commitment schemes are a useful cryptographic tool that allows to commit to a secret value with little possibility to alter it afterward. We hereby focus on a popular instantiation, Pedersen vector commitment scheme under the DLOG assumption. This scheme is:

- **Perfectly Hiding:** Even an adversary with unlimited computational power cannot extract any information about the committed values.
- **Computationally Binding:** For adversaries with bounded computational resources, the probability of opening a commitment to two different values is negligible.

Definition 2 (Pedersen Vector Commitment) *Given the message space $M = \mathbb{Z}_p^N$, the randomness space $R = \mathbb{Z}_p^*$, the commitment space $C = \mathbb{G}$ of prime order p and $(g_1,...,g_N,h) \xleftarrow{\$} \mathbb{G}$, a commitment to a message vector $(v_1,...,v_N) \leftarrow \mathbb{Z}_p^N$ is defined as:*

$$\mathrm{Com}(v_1,...,v_N;r) \triangleq \prod_{i=1}^N g_i^{v_i} \cdot h^r$$

where Pedersen commitment is a special case where $N=1$.

4.4 Zero-Knowledge Arguments of Knowledge

In this paper, we focus on zero-knowledge arguments of knowledge, which are zero-knowledge proof with computational soundness. This ensures that no probabilistic polynomial-time prover can deceive the verifier into accepting a false claim. It is an interactive protocol Π between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for a relation R. The protocol takes an NP public statement u and the prover's private *witness* ω. The verifier then outputs a decision on whether to accept or reject the prover's claim of knowing the witness based on their interaction. The sequence of messages exchanged in the protocol is called a *transcript*. An interactive protocol is termed *public-coin* if all of the verifier's messages are randomly generated and independent of the prover's messages. These verifier messages are also referred to as *challenges*.

Public-Coin Protocols. We exemplify an interactive public-coin protocol, where a prover $\mathcal{P}$ proves the knowledge of a private witness ω hidden in a commitment $W = g^{\omega} \cdot h^{r_\omega}$:

1. $\mathcal{P}$ sends an initial commitment $Q = g^q \cdot h^{r_q}$ to $\mathcal{V}$.
2. $\mathcal{V}$ generates a random challenge $e \xleftarrow{\$} \mathbb{Z}_p^*$.
3. $\mathcal{P}$ replies with two values $v = \omega + q \cdot e$ and $f = r_\omega + r_q \cdot e$.

$\mathcal{V}$ verifies $g^v \cdot h^f \stackrel{?}{=} W \cdot Q^e$ to accept or reject the proof.

Key Properties. A public-coin zero-knowledge argument of knowledge must satisfy three key properties:

- **completeness**, if the predicate $(u,\omega) \in \mathsf{R}$ is always true for any input.
- $(\gamma_1,\ldots,\gamma_\mu)$**-special soundness**, if there exists an efficient algorithm that, given a statement u and a $(\gamma_1,\ldots,\gamma_\mu)$-tree of accepting transcripts, outputs a witness ω for u. A $(\gamma_1,\ldots,\gamma_\mu)$-tree of accepting transcripts is a set of $\prod_{i=1}^{\mu}\gamma_i$ accepting transcripts arranged in a tree structure, where the nodes represent the prover's messages and the edges represent the verifier's challenges. Each transcript includes the messages along the path from the root to a leaf node. γ-*special soundness* is a special case of $(\gamma_1,\ldots,\gamma_\mu)$-*special soundness*, where $\mu = 1$.
- **special honest-verifier zero-knowledge (SHVZK)**, if given the challenges, there exists an efficient simulator that can always generate an indistinguishable transcript of the proof without knowledge of the witness.

Non-Interactivity. Interactive public-coin protocols can be transformed to non-interactive signatures of knowledge [6] via the well-known Fiat-Shamir transformation [12], where the random oracle [2] is modelled by a collision-resistant hash function. It suffices to compute the hash value of the initial messages and the message to be signed as the random challenge in the protocols.

5 An Analysis of Any-out-of-N proofs

5.1 Overview of Any-out-of-N proofs

5.1.1 "K-Weight" Technique

ZGSX23 employed a "K-Weight" technique to construct an Any-out-of-N proof. Given a public-key ring $(P_i = \tau^{s_i})_{i=1}^{|\mathcal{R}|}$, a prover $\mathcal{P}$ aims to prove the knowledge of a tuple of secret keys $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$. The constraint is formalized as below:

Definition 3 (K-Weight Technique) *A "K-Weight"-based Any-out-of-N proof should satisfy the following relation:*

$$\left\{ \begin{array}{l|l} \prod_{i=1}^{|\mathcal{R}|} (P_i^{y^i})^{b_i} = \tau^{\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}} & (\tau, (P_i)_{i=1}^{|\mathcal{R}|}) \in \mathbb{G} \\ \wedge\, (b_i \in \{0,1\})_{i=1}^{|\mathcal{R}|} & \begin{pmatrix} y, (s_k \neq 0)_{k=1}^{|\mathcal{S}|}, \\ (b_i)_{i=1}^{|\mathcal{R}|} \end{pmatrix} \in \mathbb{Z}_p \end{array} \right\}$$

The prover commits to a single bit vector $(b_i)_{i=1}^{|\mathcal{R}|}$, where $b_{\phi(k)} = 1,\ k \in \{1,...,|\mathcal{S}|\} \to \phi(k) \in \{1,...,|\mathcal{R}|\}$ and $\phi(k)$ indicates the index of the k-th source public key within the public-key ring. Then the prover uses it to generate a weighted sum $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$ for an arbitrary $y \xleftarrow{\$} \mathbb{Z}_p^*$ to show the knowledge of multiple secret keys $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$. Note that the Hamming weight of the bit vector is kept secret in the proof.

5.1.2 Associated RingCT Scheme

Transaction Model. Recall that cutting-edge RingCT schemes, e.g., Omniring and RingCT-3.0, use $|\mathcal{S}|$ tags in each transaction. To use an Any-out-of-N proof while preserving the privacy of the value $|\mathcal{S}|$, $|\mathcal{R}|$ tags are utilized in each transaction by using additional $(|\mathcal{R}| - |\mathcal{S}|)$ randomly group elements from $\mathbb{G}$ as decoy tags for the decoy accounts. The transaction tx would become:

$$(\mathcal{R}, \mathcal{S} = (T_i)_{i=1}^{|\mathcal{R}|}, \mathcal{T}, \tilde{q})$$

Stateless Tagging Scheme. ZGSX23 introduced a workaround to mitigate the significant costs associated with storing large amounts of tags on blockchain due to the tag proliferation. It introduced an RSA-based accumulator to enable a stateless tagging scheme, embedding all tags into a single accumulator value rather than storing them on-chain. An additional prime value, $\tilde{q}$, integrating the non-decoy tags is included in the transaction. Briefly, a prover generates a prime tag vector $(q_i = \mathcal{H}(T_i))_{i=1}^{|\mathcal{R}|}$ via a prime hash function $\mathcal{H}()$ based on the tag vector $(T_i)_{i=1}^{|\mathcal{R}|}$. The prime tag vector can be compressed to a single value $\tilde{q} = \prod_{i=1}^{|\mathcal{R}|} q_i^{y^i b_i} = \prod_{k=1}^{|\mathcal{S}|} q_{\phi(k)}^{y^{\phi(k)}}$ by using the bit vector to filter out the decoy prime tags.

Constraints. Employing Any-out-of-N proofs requires a slightly modified language $\mathcal{L}_A$ derived from $\mathcal{L}_K$, where the differences from $\mathcal{L}_K$ are highlighted in blue:

$$\mathcal{L}_A \triangleq \begin{cases} \mathsf{stmt} = \left((P_i, A_i^{\mathcal{R}})_{i=1}^{|\mathcal{R}|}, (T_i)_{i=1}^{|\mathcal{R}|}, (A_j^{\mathcal{T}})_{j=1}^{|\mathcal{T}|}, \tilde{q}\right): \\ \exists \mathsf{wit} = \left((s_{\phi(k)}, a_{\phi(k)}, r_{\phi(k)})_{k=1}^{|\mathcal{S}|}, (t_j, m_j)_{j=1}^{|\mathcal{T}|}\right) \text{ s.t.}: \\ \forall\, k \in \{1,...,|\mathcal{S}|\}: \begin{cases} P_{\phi(k)} = \tau^{s_{\phi(k)}},\ T_{\phi(k)} = \tau^{s_{\phi(k)}^{-1}}, \\ A_{\phi(k)}^{\mathcal{R}} = g^{a_{\phi(k)}} h^{r_{\phi(k)}} \end{cases} \\ \forall\, j \in \{1,...,|\mathcal{T}|\}: \left(A_j^{\mathcal{T}} = g^{t_j} h^{m_j},\ t_j \in [0, 2^{\beta} - 1]\right) \\ \sum_{k=1}^{|\mathcal{S}|} a_{\phi(k)} = \sum_{j=1}^{|\mathcal{T}|} t_j \\ \tilde{q} \text{ is well-formed and not a member in the accumulator} \end{cases}$$

This modification allows for anonymous transaction spending without disclosing the number of the source accounts. The following two zero-knowledge proofs are combined with a balance proof and range proofs to satisfy the constraints outlined in $\mathcal{L}_A$:

- An **Any-out-of-N proof**, to prove the knowledge of an arbitrary subset of secrets out of the ring accounts.
- An **RSA-based non-membership proof**, to prove that the tags do not exist in the accumulator. The prover proves that $\tilde{q}$ does not have any common divisor with the accumulator value.

Remark 1 *Note that ZGSX23 also uses reciprocal secret keys in both the source public keys and source tags, resulting in the creation of two "weighted sum" values, similar to Omniring. However, unlike Omniring, ZGSX23 does not include secret key vectors, making it exceptionally difficult to establish linkability by demonstrating the reciprocal relationship between any pair of secret keys. This issue will be discussed further in Section 6.*

5.2 Limitations

We enumerate several limitations that were not given adequate considerations in the original paper.

5.2.1 Increased Transaction Size

From the complexity comparison in Table 3, we can see a clear transaction size increase of ZGSX23's RingCT scheme compared to previous ones due to the two major reasons:

Tag Proliferation. Recall that using an Any-out-of-N proof requires using $|\mathcal{R}|$ tags as $|\mathcal{S}|$ must remain secret and tends to be far smaller than $|\mathcal{R}|$. Hiding multiple source accounts while preserving a reasonable level of anonymity requires a decent-sized ring, such as $|\mathcal{R}| \geq 32$. However, this causes the linear-sized tags to start to dominate the overall transaction size. As $|\mathcal{R}|$ increases, its impact on transaction size becomes more significant as evidenced by the widening gap in transaction sizes between BulletCT and ZGSX23, as shown in Figure 1. This not only negates the communication advantages offered by its compact signature but also significantly inflates the overall transaction size.

Assumption Incompatibility. Introducing an accumulator requires additional zero-knowledge proofs under different cryptographic assumptions from the DLOG one, leading to more computationally and communication-intensive operations. Consider the RSA-based accumulator used by ZGSX23. On the one hand, it is challenging to directly apply DLOG-based compression techniques to RSA-based zero-knowledge proofs, making it difficult to minimize the overall communication costs of RingCT signatures. On the other hand, RSA-based proofs typically incur higher computational and communication costs compared to DLOG-based ones because an RSA group with a minimum order of 3072 bits provides comparable security to an elliptic curve group with a 256-bit order according to the NIST recommendations[3]. Moreover, using such an accumulator requires both provers and verifiers to perform additional $|\mathcal{R}|$ computationally-intensive hash operations to generate and validate prime tags.

5.2.2 Stateless Tagging Scheme

While an accumulator offers storage benefits, the drawbacks of integrating it into tagging schemes for stateless blockchains outweigh their advantages:

Impractical Witness Update. Stateless blockchains are argued to be far from practical by a recent study (FC '23) [7]. The researchers assert that stateless blockchains require users to store additional accumulator-related witnesses[4] to help validators verify transactions. Users' witnesses may become invalid while other transactions are updating the global state. Utilizing a stateless blockchain places heavy burdens on users to actively monitor the network and refresh their witnesses periodically. Thus, the trade-off between a large global state and requiring frequent witness changes is fundamental.

Cryptographic Complexities. Handling groups under multiple cryptographic assumptions introduces additional complexities and incompatibility, resulting in potential security risks. Special care must be taken to guarantee the security under a mixture of cryptographic assumptions. Thus, minimizing the cryptographic complexities is always preferable.

Trusted Setup. A recent research [21] indicates that most efficient accumulators, such as those based on RSA or bilinear pairings, require trusted setups. Trusted setups are widely believed to pose risks of exposing secret trapdoor information, potentially compromising the security of blockchain systems. In particular, the RSA-based accumulator used by ZGSX23 involves a trusted setup for modulus generation. Moreover, ideal class groups are deemed as an optional substitute for RSA groups without using a trusted setup, which is based on the hardness of another cryptographic assumption. However, the efficiency is still incomparable to DLOG-based elliptic curve groups. According to a recent study [8], class groups of 3392-bit order can barely achieve 128-bit security as DLOG-based elliptic curve groups of 256-bit order.

5.2.3 Summary

Although using Any-out-of-N proofs achieve greater anonymity by allowing for smaller rings, these benefits are considerably outweighed by the aforementioned limitations for blockchain-based RingCT schemes. Specifically, we informally generalize that the tag proliferation is an inherent limitation of all Any-out-of-N proofs, regardless of the techniques used. This issue represents the most significant barrier preventing them from effectively addressing the scalability concerns. Therefore, our analysis indicates that Any-out-of-N proofs are less desirable for DLOG-based RingCT schemes than K-out-of-N proofs.

Remark 2 *This conclusion is drawn from the RingCT scheme developments at the time of writing. However, we do not rule out the possibility that some non-inherent limitations could be resolved in the future. For example, integrating trustless accumulators fully compatible with other DLOG-based proofs could help reduce communication overheads and eliminate the trusted setup, which is beyond the scope of this paper.*

6 New Tag Proof

Linkability is an essential property of RingCT signatures to resist double-spending attacks. Recall that we leverage "K-Weight"-based technique to develop a K-out-of-N proof by revealing the Hamming weight of the bit-vector. In this section, we elaborate on the two technical challenges and the solutions to address them, respectively.

6.1 Overview of Existing Techniques

Omniring and RingCT-3.0 conceal reciprocal secret keys in the source public keys and the tags. Their approaches use individual secret keys as witnesses so as to leverage the constraint $s \cdot s^{-1} = 1$ to demonstrate the reciprocal relations between pairs of secret keys. We briefly describe their approaches:

- **Omniring** constructs two secret key vectors $(s_k)_{k=1}^{|\mathcal{S}|}$ and $(y^k \cdot s_k^{-1})_{k=1}^{|\mathcal{S}|}$ and leverages Bulletproofs' inner product argument to prove the constraint by computing $\sum_{k=1}^{|\mathcal{S}|} s_k \cdot y^k \cdot s_k^{-1} = \sum_{k=1}^{|\mathcal{S}|} y^k$, where y is a random scalar. It further employs additional constraints to establish links between the two secret key vectors and the corresponding weighted sums derived from the public-key ring and the tags.

[3]https://www.keylength.com/en/4/

[4]This witness refers to the accumulator-based proof, which is different from the secret "witness" in zero-knowledge proofs.

- **RingCT-3.0** presents a special K-out-of-N proof, which is an aggregation of K separate 1-out-of-N proofs, where the k-th proof has an independent secret key s_k for $k \in \{1, ..., |\mathcal{S}|\}$. RingCT-3.0 uses an additional constraint $\prod_{k=1}^{|\mathcal{S}|} (\eta^{s_k^{-1}})^{s_k y^k} = \eta^{\sum_{k=1}^{|\mathcal{S}|} y^k}$ to prove the reciprocal relations for multiple secret keys.

6.2 Technical Challenge I & Solution

6.2.1 Challenge

Recall that in "K-Weight"-based proofs, the prover must demonstrate the knowledge of two weighted sums $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$ and $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}^{-1}$. The idea, however, presents an inherent challenge in directly applying the constraint $s \cdot s^{-1} = 1$ to establish separate reciprocal relations of multiple secret keys. Additionally, we discovered a linkability flaw in ZGSX23's scheme: it also use reciprocal secret keys in the source public keys and tags but fails to provide a tag proof to establish the reciprocal relations. One possible approach is to adopt Omniring's method, which uses additional secret key vectors to establish computationally intensive reciprocal constraints. However, this would significantly compromise the computational efficiency that ZGSX23 offers.

6.2.2 Solution

To address the challenge, we adopt a simpler strategy than the reciprocal one of proving the source public keys hidden in the ring and the tags share the same weighted sum $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$. To achieve this, we use an alternative tag instantiation $T = \eta^s$, where the secret key is shared with the corresponding source public key on a different generator. This instantiation has the same *pseudo-randomness* and *one-wayness* as that of Omniring (Please see Section C.4 for more details). Recall that the prover uses a bit vector to conduct filtering on the public-key ring and proves the knowledge of a weighted sum. Our intuition is that, to construct the same weighted sum from the tags, the prover must raise the tag hiding the secret key $s_{\phi(k)}$ to the power of the random challenge $y^{\phi(k)}$ for $k \in \{1, ..., |\mathcal{S}|\}$. Thus, without considering zero-knowledge, we design a core equality in Eqn. (1), where $d \xleftarrow{\$} \mathbb{Z}_p^*$ is a new random challenge to separate the generator τ from η:

$$\prod_{i=1}^{|\mathcal{R}|} P_i^{y^i b_i} \cdot \prod_{k=1}^{|\mathcal{S}|} (T_k^d)^{y^{\phi(k)}} = (\tau \cdot \eta^d)^{\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}} \tag{1}$$

In the following, we refer to the sequence of values to be raised to the power of the tags as the "tag exponent vector".

6.2.3 Patching ZGSX23's Flaw

Our solution also addresses ZGSX23's linkability flaw. With this tag instantiation and the new strategy, thanks to the equal size of the public-key ring and tag ring, the prover can easily prove that the public keys and tags of the source accounts hidden in the two rings share the same weighted sum by computing the following:

$$\prod_{i=1}^{|\mathcal{R}|} (P_i \cdot T_i^d)^{y^i b_i} = (\tau \cdot \eta^d)^{\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}} \tag{2}$$

Remark 3 *Unfortunately, the patch in Eqn. (2) is incompatible with "K-Weight"-based K-out-of-N proofs due to the mismatch in sizes between the public-key ring and tags. As a result, we developed an alternative approach, shown in Eqn. (1), to address this challenge.*

6.3 Technical Challenge II & Solution

6.3.1 Challenge

In public-coin protocols, the elements in the tag exponent vector in Eqn. (1) are secret witnesses and must be hidden in a commitment to preserve zero-knowledge and prevent malicious provers from arbitrarily altering them. Then the prover provides a sequence of masking values hiding the elements of the tag exponent vector for verification. However, a more challenging issue arises: the proof is not sound as it lacks a constraint that ensures the tag exponent vector is indeed $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$. Let us describe how a malicious prover can cheat without considering zero-knowledge: the prover prepares invalid tags $(T_k = \eta^{s'_k})_{k=1}^{|\mathcal{S}|}$ with the secret key sequence $(s'_k)_{k=1}^{|\mathcal{S}|}$ totally different from the secret key sequence $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$ embedded in the source public keys before receiving the random value y from the verifier. The prover can easily use a randomly sampled tag exponent vector $(\alpha_k)_{k=1}^{|\mathcal{S}|}$ rather than $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$ to generate an equal weighted sum $\sum_{k=1}^{|\mathcal{S}|} \alpha_k s'_k$ as the value $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$ extracted from the public-key ring. The consequence is serious as this would expose RingCT schemes to double-spending attacks.

6.3.2 Solution

With only a single weighted sum $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}$, it becomes particularly challenging to restrict the elements of the tag exponent vector to specific values $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$ while retaining zero-knowledge property. To address this critical challenge, we propose an elegant, indirect solution: we introduce a new constraint on the tag exponent vector, requiring it to be a permutation of the sequence $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$ to ensure that the two secret key sequences satisfy a permutation relation. Recall that to enable linkability to resist double-spending attacks, it suffices to ensure each revealed tag in a transaction is uniquely associated with one of the source public keys and has not been previously revealed. Thus, establishing a bijection in a zero-knowledge manner between the two secret key sequences hidden in the source public keys and tags in arbitrary order, rather than an order-preserving bijection as used in Omniring, is adequate. We formalize an important lemma that underpins our solution:

Lemma 1 *Given two sequences of secret keys* $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$ *and* $(s'_k)_{k=1}^{|\mathcal{S}|}$ *hidden in the source public keys and tags, for an arbitrary value* $y \xleftarrow{\$} \mathbb{Z}_p^*$*, if the tag exponent vector* $(\alpha_k)_{k=1}^{|\mathcal{S}|}$ *satisfies a "weighted sum" constraint* $\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)} = \sum_{k=1}^{|\mathcal{S}|} \alpha_k s'_k$ *and a permutation constraint* $(\alpha_k)_{k=1}^{|\mathcal{S}|} \sim (y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$*, the probability that the two secret key sequences fail to satisfy a permutation relation is negligible for all PPT adversaries* $\mathcal{A}$*:*

$$Pr\left[\begin{array}{l|l} \sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)} = \sum_{k=1}^{|\mathcal{S}|} \alpha_k s'_k, & \mathsf{ck} \leftarrow \mathcal{G}(\lambda), \\ (\alpha_k)_{k=1}^{|\mathcal{S}|} \sim (y^{\phi(k)})_{k=1}^{|\mathcal{S}|}, & \mathcal{A}(\mathsf{ck}) \rightarrow \\ (s_{\phi(k)})_{k=1}^{|\mathcal{S}|} \nsim (s'_k)_{k=1}^{|\mathcal{S}|} & \begin{pmatrix} s_{\phi(k)} \\ s'_k \end{pmatrix}_{k=1}^{|\mathcal{S}|} \end{array}\right] \leq \mathsf{negl}(\lambda)$$

where $\sim$ and $\nsim$ denote the permutation and non-permutation operator, respectively.

Proof. Assume a PPT adversary $\mathcal{A}$ embeds two secret key sequences into the public keys and tags, which do not satisfy permutation relations. Then $\mathcal{A}$ sees the random challenge y and manages to create a tag exponent vector $(\alpha_k)_{k=1}^{|\mathcal{S}|}$ that satisfies both the "weighted sum" and permutation constraints. Given the uniformly randomness of y within the large space $\mathbb{Z}_p^*$ and by the Schwartz-Zippel lemma, $\mathcal{A}$ cannot create such two secret key sequences with overwhelming probability unless she is able to break the binding property of public keys and tags in the DLOG setting.

6.3.3 Instantiation

Based on Lemma 1, we utilize the bit vector from the K-out-of-N proof to establish a specialized permutation equality. We instantiate this equality using the well-established *logarithmic derivative* technique, introduced in a recent work (Eurocrypt '24) [9]. This technique, widely used for constructing lookup proofs, enables proving that multiple elements originate from a public set. We borrow a formalized lemma of the logarithmic derivative, namely, *Set Inclusion* from the study [13]:

Lemma 2 (Set Inclusion) *Given a sequence of field elements* $(\theta_i)_{i=1}^{N} \in \mathbb{Z}_p$ *and its deduplicated version* $(\chi_j)_{j=1}^{J} \in \mathbb{Z}_p$*, there exists a sequence of field elements* $(m_j)_{j=1}^{J} \in \mathbb{Z}_p$ *that holds for an arbitrary value* $x \xleftarrow{\$} \mathbb{Z}_p^*$*:*

$$\sum_{i=1}^{N} \frac{1}{\theta_i + x} = \sum_{j=1}^{J} \frac{m_j}{\chi_j + x} \tag{3}$$

where m_j is the *multiplicity* of the element χ_j in $(\theta_i)_{i=1}^{N}$. In our case, to prove the permutation relation between the tag exponent vector $(\alpha_k)_{k=1}^{|\mathcal{S}|}$ and the sequence $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$, we leverage the bit vector $(b_i)_{i=1}^{|\mathcal{R}|}$ to create the equality based on Eqn. (3), where all the multiplicity values are set to 1:

$$\sum_{i=1}^{|\mathcal{R}|} \frac{b_i}{y^i + x} = \sum_{k=1}^{|\mathcal{S}|} \frac{1}{y^{\phi(k)} + x} = \sum_{k=1}^{|\mathcal{S}|} \frac{1}{\alpha_k + x} \tag{4}$$

7 Instantiations

Constructing a RingCT signature requires careful consideration of seamless integration with its core component, zero-knowledge range proofs. Several blockchain-friendly range proofs have been proposed, with notable examples including Bulletproofs [5], Flashproofs [22], SwiftRange [23], and FlashSwift [17]. In this work, we build on Bulletproofs' techniques to develop compact instantiations of our "K-Weight"-based K-out-of-N proof and tag proof, ensuring seamless integration with Bulletproofs to create a compact and efficient new RingCT signature. Alternative instantiations can also be constructed based on our core ideas. We start with a brief overview to provide insights into Bulletproofs' techniques and the instantiated Any-out-of-N proof.

7.1 Technical Overview

7.1.1 Bulletproofs' Techniques

Bulletproofs introduced an inner product protocol that enables a prover to demonstrate knowledge of the inner product of two committed vectors. This approach embeds the necessary constraints into an inner product relation, which is then recursively compressed to logarithmic size. The full protocol of Bulletproofs is the composition of $\log\beta$ times of inner product protocols Π_{IP} and a range protocol Π_{RG} in Eqn. (5).

$$\Pi_{\mathsf{full}} = \underbrace{\Pi_{\mathsf{IP}} \diamond \cdots \diamond \Pi_{\mathsf{IP}}}_{\log\beta \text{ times}} \diamond \Pi_{\mathsf{RG}} \tag{5}$$

Bit-Decomposition Technique. In the range protocol Π_{RG}, Bulletproofs proves that a committed value v lies within the range $[0, 2^\beta - 1]$ by demonstrating the existence of a β-bit vector $\mathbf{b}$ that constitutes the value v. The protocol achieves this by combining three constraints, separated by three scalars $(1, z, z^2)$, where $y, z \xleftarrow{\$} \mathbb{Z}_p^*$, $\mathbf{1}^\beta = (1, ..., 1)$, $\mathbf{2}^\beta = (2^{i-1})_{i=1}^{\beta}$ and $\mathbf{y}^\beta = (y^{i-1})_{i=1}^{\beta}$.

$$\langle \mathbf{y}^\beta, \mathbf{b} \cdot \mathbf{a} \rangle + z \cdot \langle \mathbf{y}^\beta, \mathbf{b} - \mathbf{1}^\beta - \mathbf{a} \rangle + z^2 \cdot \langle \mathbf{2}^\beta, \mathbf{b} \rangle = z^2 \cdot v \tag{6}$$

The first and second constraints aim to prove the two committed vectors $\mathbf{b}$ and $\mathbf{a} = \mathbf{1}^\beta - \mathbf{b}$ are binary ones while the third one aims to prove that v can be written as the inner product $\langle \mathbf{2}^\beta, \mathbf{b} \rangle$.

Inner Product Protocol. Bulletproofs employs an inner product protocol Π_{IP} to enforce the constraint in Eqn. (6) by constructing an inner product expression $Q = \mathbf{g}^{\mathbf{b}} \cdot \mathbf{h}^{\mathbf{a}} \cdot \tau^{\langle \mathbf{b}, \mathbf{a} \rangle}$, where Q is a group element. The inner product protocol Π_{IP} can be recursively compressed to achieve $\log \beta$ communication complexity by trading half number of the two scalar vectors for additional two group elements until the vector dimension β is reduced to 1. Please see the full protocol in Appendix A.

Multi-Exponentiation Technique. Bulletproofs also employs an established multi-exponentiation technique [18] to improve verifier efficiency for optimization. Using the multi-exponentiation technique can reduce the verification to a single multi-exponentiation by delaying all the group exponentiations to the last round. The verifier can aggregate the exponents of the generators before performing one-off exponentiations for verification.

Performance. Bulletproofs involves $2\log \beta + 9$ elements. By using the multi-exponentiation technique, its verification is dominated by 2β group exponentiations for employing 2β distinct generators to commit to the vectors $\mathbf{b}$ and $\mathbf{a}$.

7.1.2 Any-out-of-N Proof

The Any-out-of-N proof relies solely on the first two constraints to demonstrate the existence of an N-bit vector $\mathbf{b}$ without disclosing its Hamming weight:

$$\langle \mathbf{y}^N, \mathbf{b} \cdot \mathbf{a} \rangle + z \cdot \langle \mathbf{y}^N, \mathbf{b} - \mathbf{1}^N - \mathbf{a} \rangle = 0 \tag{7}$$

The Any-out-of-N proof inherits Bulletproofs' performance. Its verification is dominated by $3N$ group exponentiations, including $2N$ group exponentiations arising from the two vectors ($\mathbf{b}$ and $\mathbf{a}$) and an additional N group exponentiations associated with the public-key ring.

7.2 Our K-out-of-N proof

Similar to the Any-out-of-N proof, our proof uses Bulletproofs' bit-decomposition technique to prove the existence of a valid bit-vector whose Hamming weight is equal to the public value $|\mathcal{S}|$. Then the prover uses the bit vector to prove the knowledge of the weighted sum of a subset of secret keys. We present a slightly adapted equation based on the one used in Bulletproofs in Eqn. (6), which forms the foundation of our protocol:

$$\langle \mathbf{y}^{|\mathcal{R}|}, \mathbf{b} \cdot \mathbf{a} \rangle + z \cdot \langle \mathbf{y}^{|\mathcal{R}|}, \mathbf{b} - \mathbf{1}^{|\mathcal{R}|} - \mathbf{a} \rangle + z^2 \cdot \langle \mathbf{1}^{|\mathcal{R}|}, \mathbf{b} \rangle = z^2 \cdot |\mathcal{S}| \tag{8}$$

where we replace the committed value v in Bulletproofs with the public value $|\mathcal{S}|$.

7.2.1 Full Protocol

Given public parameters $\tau, g, h, (g_i)_{i=1}^{|\mathcal{R}|}, (h_i)_{i=1}^{|\mathcal{R}|} \xleftarrow{\$} \mathbb{G}$, public inputs $(P_i)_{i=1}^{|\mathcal{R}|} \in \mathbb{G}$ and secret witnesses $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$, $(b_i)_{i=1}^{|\mathcal{R}|}$, $(a_i = 1 - b_i)_{i=1}^{|\mathcal{R}|} \in \mathbb{Z}_p$, where $(P_{\phi(k)} = \tau^{s_{\phi(k)}})_{k=1}^{|\mathcal{S}|}$, our full protocol goes as below:

$$\mathcal{P} : (r_{u_i})_{i=1}^{|\mathcal{R}|}, (r_{n_i})_{i=1}^{|\mathcal{R}|}, r_b, r_c, r_w, r_{t_1}, r_{t_2} \xleftarrow{\$} \mathbb{Z}_p^* \tag{9}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : B \triangleq \prod_{i=1}^{|\mathcal{R}|} g_i^{b_i} \cdot h_i^{a_i} \cdot h^{r_b} \tag{10}$$

$$\mathcal{P} \Leftarrow \mathcal{V} : y \xleftarrow{\$} \mathbb{Z}_p^* \tag{11}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : T_2 \triangleq g^{t_2} \cdot h^{r_{t_2}} \tag{12}$$

$$\mathcal{P} \Leftarrow \mathcal{V} : z \xleftarrow{\$} \mathbb{Z}_p^* \tag{13}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : T_1 \triangleq g^{t_1} \cdot h^{r_{t_1}} \tag{14}$$

$$\mathcal{P} \Leftarrow \mathcal{V} : d \xleftarrow{\$} \mathbb{Z}_p^* \tag{15}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : C \triangleq \prod_{i=1}^{|\mathcal{R}|} (g_i \cdot P_i^{dy^i})^{r_{u_i}} \cdot h_i^{r_{n_i}} \cdot h^{r_c} \cdot \tau^{-r_w} \tag{16}$$

$$\mathcal{P} \Leftarrow \mathcal{V} : e \xleftarrow{\$} \mathbb{Z}_p^* \tag{17}$$

$$\mathcal{P} \Rightarrow \mathcal{V} : w \triangleq \sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)} + r_w \cdot e \tag{18}$$

$$\hat{t} \triangleq \langle \mathbf{u}, \mathbf{n} \rangle = z^2 |\mathcal{S}| + \delta(y,z) + t_1 \cdot e + t_2 \cdot e^2 \tag{19}$$

$$\mathbf{u} \triangleq (u_i \triangleq b_i - z + r_{u_i} \cdot e)_{i=1}^{|\mathcal{R}|} \tag{20}$$

$$\mathbf{n} \triangleq \left(n_i \triangleq y^i \cdot (a_i + z + r_{n_i} \cdot e)\right)_{i=1}^{|\mathcal{R}|} \tag{21}$$

$$r_{\hat{t}} \triangleq r_{t_2} \cdot e^2 + r_{t_1} \cdot e \tag{22}$$

$$r_q \triangleq r_c \cdot e + r_b \tag{23}$$

$$\mathcal{V} : g^{\hat{t}} \cdot h^{r_{\hat{t}}} \stackrel{?}{=} g^{z^2|\mathcal{S}| + \delta(y,z)} \cdot T_1^e \cdot T_2^{e^2} \tag{24}$$

$$LHS \stackrel{?}{=} RHS \tag{25}$$

$$LHS \triangleq \prod_{i=1}^{|\mathcal{R}|} (g_i \cdot P_i^{dy^i})^{u_i} \cdot (h_i^{y^{-i}})^{n_i} \cdot h^{r_q} \tag{26}$$

$$RHS \triangleq \tau^{wd} \cdot B \cdot C^e \cdot \prod_{i=1}^{|\mathcal{R}|} (g_i \cdot P_i^{dy^i})^{-z} \cdot h_i^z \tag{27}$$

where $t_1 \triangleq \sum_{i=1}^{|\mathcal{R}|} y^i \cdot \left(r_{n_i} \cdot (b_i - z) + r_{u_i} \cdot (a_i + z)\right)$, $t_2 \triangleq \sum_{i=1}^{|\mathcal{R}|} y^i \cdot r_{u_i} \cdot r_{n_i}$ and $\delta(y,z) \triangleq (z - z^2) \cdot \sum_{i=1}^{|\mathcal{R}|} y^i - z^3$.

7.2.2 Highlights

Compared to the Any-out-of-N protocol, we make two major adaptations as below:

- We shift the challenge vector $(y^i)_{i=1}^{|\mathcal{R}|}$ from $\mathbf{u}$ to $\mathbf{n}$, aligning it with the vector structure of Bulletproofs and facilitating seamless integration with Bulletproofs. More importantly, this adaptation facilitates the construction of our tag proof, which we will discuss in Section 7.3.

- We reduce the number of elements by one at the cost of an extra round with an additional random challenge d, without compromising soundness.

Theorem 1 *Our K-out-of-N proof has perfect completeness, computational witness extended emulation and perfect special honest-verifier zero-knowledge (SHVZK).*

The proof for Theorem 1 is given in Appendix C.2.

7.3 Our Tag Proof

Our tag proof aims to ensure the linkability by demonstrating that the same weighted sum can be derived from both the public keys and tags even when they are of unequal sizes. Furthermore, according to Lemma 1, the prover must prove that the tag exponent vector is a permutation of the sequence $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$.

7.3.1 Full Protocol

Given the masking value w and the valid vector $(u_i)_{i=1}^{|\mathcal{R}|}$ hiding the weighted sum and the bit vector $(b_i)_{i=1}^{|\mathcal{R}|}$ from our K-out-of-N proof, public parameters $\eta, g, h, (g_k)_{k=1}^{|\mathcal{S}|}, (h_k)_{k=1}^{|\mathcal{S}|} \xleftarrow{\$} \mathbb{G}$, public inputs $(T_k)_{k=1}^{|\mathcal{S}|} \in \mathbb{G}$, and secret witnesses $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|} \in \mathbb{Z}_p$, our full protocol goes as follows:

$$\mathcal{P} : (r_{v_k})_{k=1}^{|\mathcal{S}|}, (r_{f_k})_{k=1}^{|\mathcal{S}|}, r_y, r_e, r_z, r_{m_1}, r_{m_2} \xleftarrow{\$} \mathbb{Z}_p^* \quad (28)$$

$$\mathcal{P} \Leftarrow \mathcal{V} : y \xleftarrow{\$} \mathbb{Z}_p^* \quad (29)$$

$$\mathcal{P} \Rightarrow \mathcal{V} : Y \triangleq \prod_{k=1}^{|\mathcal{S}|} g_k^{y^{\phi(k)}} \cdot h^{r_y} \quad (30)$$

$$\mathcal{P} \Leftarrow \mathcal{V} : z \xleftarrow{\$} \mathbb{Z}_p^* \quad (31)$$

$$\mathcal{P} \Rightarrow \mathcal{V} : Z \triangleq \prod_{k=1}^{|\mathcal{S}|} h_k^{(y^{\phi(k)}+z)^{-1}} \cdot h^{r_z} \quad (32)$$

$$\mathcal{P} \Leftarrow \mathcal{V} : c \xleftarrow{\$} \mathbb{Z}_p^* \quad (33)$$

$$\mathcal{P} \Rightarrow \mathcal{V} : M_1 \triangleq g^{m_1} \cdot h^{r_{m_1}}, \quad M_2 \triangleq g^{m_2} \cdot h^{r_{m_2}} \quad (34)$$

$$\mathcal{P} \Leftarrow \mathcal{V} : d \xleftarrow{\$} \mathbb{Z}_p^* \quad (35)$$

$$\mathcal{P} \Rightarrow \mathcal{V} : E \quad (36)$$

$$\mathcal{P} \Leftarrow \mathcal{V} : e \xleftarrow{\$} \mathbb{Z}_p^* \quad (37)$$

$$\mathcal{P} \Rightarrow \mathcal{V} : \hat{m} \triangleq \langle \mathbf{v}, \mathbf{f} \rangle = \sum_{k=1}^{|\mathcal{S}|} c^k + m_1 \cdot e + m_2 \cdot e^2 \quad (38)$$

$$\mathbf{v} \triangleq \left(v_k \triangleq c^k \cdot (y^{\phi(k)} + z + r_{v_k} \cdot e)\right)_{k=1}^{|\mathcal{S}|} \quad (39)$$

$$\mathbf{f} \triangleq (f_k \triangleq (y^{\phi(k)} + z)^{-1} + r_{f_k} \cdot e)_{k=1}^{|\mathcal{S}|} \quad (40)$$

$$r_{\hat{m}} \triangleq r_{m_2} \cdot e^2 + r_{m_1} \cdot e \quad (41)$$

$$r_s \triangleq r_y + r_e \cdot e + r_z \cdot d \quad (42)$$

$$\mathcal{V} : g^{\hat{m}} \cdot h^{r_{\hat{m}}} \stackrel{?}{=} g^{\sum_{k=1}^{|\mathcal{S}|} c^k} \cdot M_1^e \cdot M_2^{e^2} \quad (43)$$

$$LHS \stackrel{?}{=} RHS \quad (44)$$

where $m_1 \triangleq \sum_{k=1}^{|\mathcal{S}|} c^k \cdot \left(r_{f_k} \cdot (y^{\phi(k)} + z) + r_{v_k} \cdot (y^{\phi(k)} + z)^{-1}\right)$, $m_2 \triangleq \sum_{k=1}^{|\mathcal{S}|} c^k \cdot r_{v_k} \cdot r_{f_k}$ and:

$$\begin{aligned} E \triangleq & \prod_{k=1}^{|\mathcal{S}|} (g_k \cdot T_k^{d^2})^{r_{v_k}} \cdot h_k^{d r_{f_k}} \cdot h^{r_e} \cdot \eta^{-d^2 \cdot r_w} \\ & \cdot \epsilon^{d^3 \cdot \left(\sum_{k=1}^{|\mathcal{S}|} r_{f_k} - \sum_{i=1}^{|\mathcal{R}|} r_{u_i} \cdot (y^i + z)^{-1}\right)} \end{aligned} \quad (45)$$

$$\begin{aligned} LHS \triangleq & \prod_{i=1}^{|\mathcal{R}|} (\epsilon^{-d^3 (y^i+z)^{-1}})^{u_i} \cdot \prod_{k=1}^{|\mathcal{S}|} \left((g_k \cdot T_k^{d^2})^{c^{-k}}\right)^{v_k} \\ & \cdot \prod_{k=1}^{|\mathcal{S}|} (h_k^d \cdot \epsilon^{d^3})^{f_k} \cdot h^{r_s} \end{aligned} \quad (46)$$

$$\begin{aligned} RHS \triangleq & \eta^{d^2 \cdot w} \cdot (\prod_{k=1}^{|\mathcal{S}|} T_k)^{d^2 \cdot z} \cdot \epsilon^{d^3 z \sum_{i=1}^{|\mathcal{R}|} (y^i+z)^{-1}} \\ & \cdot Y \cdot (\prod_{k=1}^{|\mathcal{S}|} g_k)^z \cdot Z^d \cdot E^e \end{aligned} \quad (47)$$

7.3.2 Verification Dissection

We provide a detailed analysis of our verification equations to offer insights into the underlying intuition and to demonstrate the *completeness* property of our proof. Please see Theorem 2 and its security proof for a more detailed analysis of the *soundness* and *zero-knowledge* properties.

- Firstly, the following two equations give us the linear forms of $\left(v_k \triangleq c^k \cdot (y^{\phi(k)} + z + r_{v_k} \cdot e)\right)_{k=1}^{|\mathcal{S}|}$ and $\left(f_k \triangleq (y^{\phi(k)} + z)^{-1} + r_{f_k} \cdot e\right)_{k=1}^{|\mathcal{S}|}$ in the challenge e, respectively.

$$\prod_{k=1}^{|\mathcal{S}|} g_k^{c^{-k} v_k} = Y \cdot \prod_{k=1}^{|\mathcal{S}|} g_k^z \cdot (\underbrace{\prod_{k=1}^{|\mathcal{S}|} g_k^{r_{v_k}}}_{\text{in } E})^e \quad (48)$$

$$\prod_{k=1}^{|\mathcal{S}|} h_k^{f_k} = Z \cdot (\underbrace{\prod_{k=1}^{|\mathcal{S}|} h_k^{r_{f_k}}}_{\text{in } E})^e \quad (49)$$

- Given two challenges c and z, this equation indicates that the two intercept values $c^k \cdot (y^{\phi(k)} + z)$ and $(y^{\phi(k)} + z)^{-1}$ of the k-th values v_k and f_k satisfy a reciprocal relation.

$$\langle \mathbf{v}, \mathbf{f} \rangle = \sum_{k=1}^{|\mathcal{S}|} c^k + m_1 \cdot e + m_2 \cdot e^2 \quad (50)$$

- By integrating the vector $\left(f_k \triangleq (y^{\phi(k)}+z)^{-1}+r_{f_k}\cdot e\right)_{k=1}^{|\mathcal{S}|}$ and the valid vector $(u_i = b_i - z + r_{u_i}\cdot e)_{i=1}^{|\mathcal{R}|}$ from our K-out-of-N proof, the following equation enforces the constraint in our permutation equality in Eqn. (4).

$$\epsilon^{\sum_{k=1}^{|\mathcal{S}|} f_k - \sum_{i=1}^{|\mathcal{R}|} u_i\cdot(y^i+z)^{-1}} = \epsilon^{z\sum_{i=1}^{|\mathcal{R}|}(y^i+z)^{-1}} \cdot \underbrace{(\epsilon^{\sum_{k=1}^{|\mathcal{S}|} r_{f_k} - \sum_{i=1}^{|\mathcal{R}|} r_{u_i}\cdot(y^i+z)^{-1}})^e}_{\text{in } E} \quad (51)$$

- Finally, given the vector $(v_k \triangleq c^k\cdot(y^{\phi(k)}+z+r_{v_k}\cdot e)_{k=1}^{|\mathcal{S}|}$, this equation shows that the same weighted sum hidden in w can also be established from the tags $(T_k)_{k=1}^{|\mathcal{S}|}$:

$$\prod_{k=1}^{|\mathcal{S}|}(T_k^{c^{-k}})^{v_k} = \eta^w\cdot(\prod_{k=1}^{|\mathcal{S}|}T_k)^z\cdot\underbrace{(\prod_{k=1}^{|\mathcal{S}|}T_k^{r_{v_k}}\cdot\eta^{-r_w})^e}_{\text{in } E} \quad (52)$$

Remark 4 *Note that the verification equation in Eqn. (43) is derived from the equation in Eqn. (50) whereas the verification equation (44) is a combination of four equations in Eqn. (48), (49), (51) and (52).*

7.3.3 Highlights

We applied three elegant tricks to our protocol:

- Our tag proof needs to use the vector $\mathbf{u}$ of our K-out-of-N proof that hides the bit-vector $\mathbf{b}$. Recall that, in K-out-of-N proof, we shift the challenge vector $(y^i)_{i=1}^{|\mathcal{R}|}$ from $\mathbf{u}$ to $\mathbf{n}$ to properly construct the permutation equality $\prod_{i=1}^{|\mathcal{R}|}(\epsilon^{(y^i+z)^{-1}})^{b_i} = \prod_{k=1}^{|\mathcal{S}|}\epsilon^{(y^{\phi(k)}+z)^{-1}}$. Without this shift, $(y^i)_{i=1}^{|\mathcal{R}|}$ would appear on the left-hand side, making it impossible construct the desired equality.
- Establishing the permutation equality in Eqn. (4) requires the prover to construct a vector $\left((y^{\phi(k)}+z)^{-1}\right)_{k=1}^{|\mathcal{S}|}$. However, a challenging issue arises in establishing the connection between the two committed vectors $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$ and $\left((y^{\phi(k)}+z)^{-1}\right)_{k=1}^{|\mathcal{S}|}$. To address this issue, we utilize the inner product protocol to implement a reciprocal constraint $(y^{\phi(k)}+z)\cdot(y^{\phi(k)}+z)^{-1} = 1$ in Eqn. (43).
- To ensure that the two masking vectors $\mathbf{v}$ and $\mathbf{f}$, concealing the witness vectors $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$ and $\left((y^{\phi(k)}+z)^{-1}\right)_{k=1}^{|\mathcal{S}|}$, satisfy the linear forms in Eqn. (39) and (40), respectively, we use two commitments, Y and Z, rather than just one, as is typical in other Bulletproofs-based proofs. In Eqn. (43), we raise the generator vector $(h_k)_{k=1}^{|\mathcal{S}|}$ to a different challenge d to distinguish it from the generator vector $(g_k)_{k=1}^{|\mathcal{S}|}$. This approach ensures that the two generator vectors are separately involved in the commitments Y and Z in Eqn. (44).

Theorem 2 *Given the value w and the valid vector $\mathbf{u}$ hiding the weighted sum and the bit vector $(b_i)_{i=1}^{|\mathcal{R}|}$ from the K-out-of-N proof, respectively, our tag proof has perfect completeness, computational witness extended emulation and perfect special honest-verifier zero-knowledge (SHVZK).*

The proof for Theorem 2 is given in Appendix C.3.

7.4 BulletCT

To construct the full RingCT signature for BulletCT, we combine the four types of proofs, *K-out-of-N proof*, *tag proof*, *balance proof* and *range proofs*, together based on the folklore observation that checking $g^\alpha = Q_1$ and $g^\theta = Q_2$ amounts to checking $g^{\alpha x+\theta} = Q_1^x\cdot Q_2$ for an arbitrary $x \xleftarrow{\$} \mathbb{Z}_p^*$. For example, given two equations $\mathbf{g}_1\cdot\mathbf{h}_1\cdot h^{\langle\mathbf{b}_1,\mathbf{a}_1\rangle} = Q_1$ and $\mathbf{g}_2\cdot\mathbf{h}_2\cdot h^{\langle\mathbf{b}_2,\mathbf{a}_2\rangle} = Q_2$, we can concatenate them by using a new challenge $x \xleftarrow{\$} \mathbb{Z}_p^*$:

$$(\mathbf{g}_1^{\mathbf{b}_1}\cdot\mathbf{h}_1^{\mathbf{a}_1}\cdot h^{\langle\mathbf{b}_1,\mathbf{a}_1\rangle})\cdot(\mathbf{g}_2^{\mathbf{b}_2}\cdot\mathbf{h}_2^{\mathbf{a}_2}\cdot h^{\langle\mathbf{b}_2,\mathbf{a}_2\rangle})^x = Q_1\cdot Q_2^x \quad (53)$$

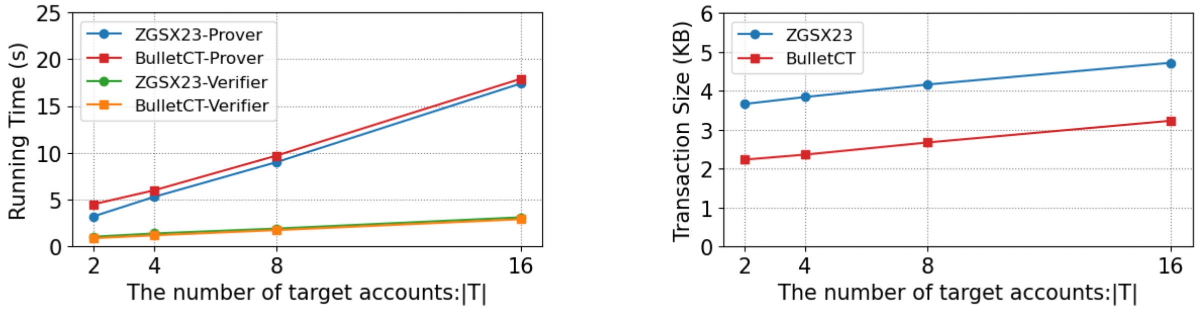
We can then transform the left-hand side of the concatenated equation into the compression-friendly form of Π_{IP} for a holistic compression, under the condition that $\mathbf{g}_1, \mathbf{g}_2, \mathbf{h}_1, \mathbf{h}_2$, and h are all distinct generators with non-trivial secret DLOG relations among them.

$$\begin{aligned}
&(\mathbf{g}_1^{\mathbf{b}_1}\cdot\mathbf{h}_1^{\mathbf{a}_1}\cdot h^{\langle\mathbf{b}_1,\mathbf{a}_1\rangle})\cdot(\mathbf{g}_2^{\mathbf{b}_2}\cdot\mathbf{h}_2^{\mathbf{a}_2}\cdot h^{\langle\mathbf{b}_2,\mathbf{a}_2\rangle})^x\\
=\ &(\mathbf{g}_1^{\mathbf{b}_1}\cdot\mathbf{h}_1^{\mathbf{a}_1}\cdot h^{\langle\mathbf{b}_1,\mathbf{a}_1\rangle})\cdot(\mathbf{g}_2^{x\mathbf{b}_2}\cdot\mathbf{h}_2^{x\mathbf{a}_2}\cdot h^{\langle x\mathbf{b}_2,\mathbf{a}_2\rangle})\\
=\ &\underbrace{\mathbf{g}_1^{\mathbf{b}_1}\cdot\mathbf{g}_2^{(x\mathbf{b}_2)}}_{=\mathbf{g}^{\mathbf{b}}}\cdot\underbrace{\mathbf{h}_1^{\mathbf{a}_1}\cdot(\mathbf{h}_2^x)^{\mathbf{a}_2}}_{=\mathbf{h}^{\mathbf{a}}}\cdot h^{\langle\mathbf{b}_1,\mathbf{a}_1\rangle+\langle x\mathbf{b}_2,\mathbf{a}_2\rangle=\langle\mathbf{b},\mathbf{a}\rangle}
\end{aligned} \quad (54)$$

where the two witness vectors and their inner product are redefined as $\mathbf{b} = \mathbf{b}_1 \mid\mid x\mathbf{b}_2$, $\mathbf{a} = \mathbf{a}_1 \mid\mid \mathbf{a}_2$ and $\langle\mathbf{b},\mathbf{a}\rangle = \langle\mathbf{b}_1,\mathbf{a}_1\rangle + \langle x\mathbf{b}_2,\mathbf{a}_2\rangle$, respectively. The generators are updated to $\mathbf{g} = \mathbf{g}_1 \mid\mid \mathbf{g}_2$ and $\mathbf{h} = \mathbf{h}_1 \mid\mid \mathbf{h}_2^x$.

8 Experimental Evaluation

Beyond the efficiency comparisons in Section 3.2.4, we experimentally benchmarked the computational costs of our scheme against the patched ZGSX23 as a baseline while the communication costs can be directly calculated based on Table 3. We employed the standard elliptic curve group on Ethereum, *BN-128* for the Pedersen commitment schemes and the common range size $\beta = 64$ for range proofs. Furthermore, we used the well-known Bouncy Castle Crypto APIs [4] to implement the BN-128 elliptic curve. All the experiments were executed on the JRE 11 in a single thread with an Apple M1 Pro processor. Note that the Java implementations are only aimed for performance comparison while some low-level programming languages, e.g., Rust and C, are more practical options. Our code is available at the Zenodo repository.



(a) The running time of prover and verifier in seconds.

(b) The transaction sizes in kilobytes.

Figure 2: The efficiency comparisons of BulletCT and the patched ZGSX23 for the ringset-II with varying number of target accounts at a comparable level of anonymity.

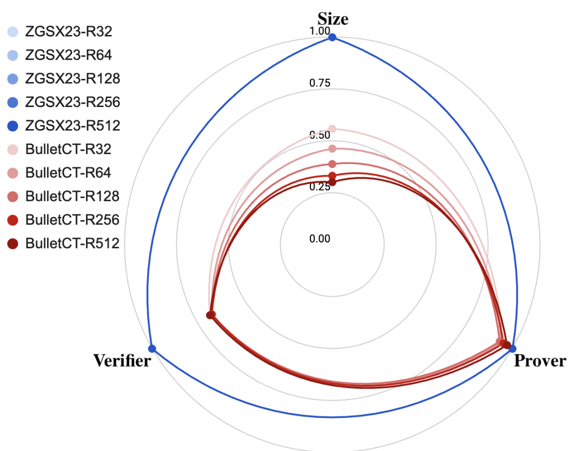


Figure 3: The normalized efficiency comparisons of BulletCT and the patched ZGSX23 with varying ring sizes $|\mathcal{R}| \in \{32, 64, 128, 256, 512\}$, where the closer the vertices to the center, the higher efficiency.

8.1 Comparisons with respect to varying $|\mathcal{T}|$

Figure 2a and 2b show comparisons of running time in seconds and transaction sizes in kilobytes with increased number of target accounts $|\mathcal{T}|$ at a comparable level of anonymity for a fair comparison, respectively. From Figure 2a, it can be observed that BulletCT and the patched ZGSX23 have comparable computational costs: our prover runs slightly slower, while our verifier is slightly faster than those of ZGSX23. Our experimental results align with the complexity comparisons shown in Table 3, indicating that the computational costs are primarily dominated by the group exponentiations. Additionally, as shown in Figure 2b, the transaction sizes in our scheme range from 2.23KB to 3.23KB, achieving 1.45× to 1.65× efficiency improvements in communication costs compared to the range from 3.66KB to 4.72KB observed in ZGSX23.

8.2 Comparisons with respect to varying $|\mathcal{R}|$

In Figure 3, we present another normalized efficiency comparison in radar charts with varying ring sizes without considering anonymity levels. Specifically, we maintain the same settings of $|\mathcal{T}| = \frac{|\mathcal{R}|}{16}$ as used in ZGSX23 for an interesting comparison. Overall, our scheme, BulletCT, demonstrates more significant advantages in transaction sizes and verifier efficiency. Specifically, compared to ZGSX23, our transaction sizes are reduced to only 30% to 56%, while our verifier costs remain constant at approximately 67% of theirs. The advantage of BulletCT in communication costs becomes greater as ring sizes grow larger. Additionally, our scheme maintains a slight edge in prover efficiency as well.

9 Conclusion

In this work, we first offered an analysis of Any-out-of-N proofs and demonstrated that Any-out-of-N proofs are less desirable for DLOG-based RingCT schemes. Secondly, we constructed a "K-Weight"-based K-out-of-N proof and the first tag proof to enable the linkability of RingCT signatures built from the former. Thirdly, we presented a new DLOG-based RingCT signature by incorporating the two proofs, without using a trusted setup. Our construction provides a more scalable RingCT scheme, making it well-suited for real-world RingCT applications, e.g., Monero. Furthermore, the rise of post-quantum RingCT schemes [10, 11] underscores the need for quantum-resistant RingCT protocols, making it essential to extend BulletCT to a post-quantum setting for future research.

10 Acknowledgements

We sincerely thank the anonymous reviewers for their valuable and insightful feedback, which greatly enhanced the quality of this paper.

References

[1] ASIM, U. Time to finality (ttf): The ultimate metric for blockchain speed. https://www.avax.network/blog/time-to-finality-ttf-the-ultimate-metric-for-blockchain-speed, 2024.

[2] BELLARE, M., AND ROGAWAY, P. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (1993).

[3] BOOTLE, J., CERULLI, A., CHAIDOS, P., GROTH, J., AND PETIT, C. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *Advances in Cryptology – EUROCRYPT 2016* (2016), M. Fischlin and J.-S. Coron, Eds., Springer Berlin Heidelberg, pp. 327–357.

[4] BOUNCYCASTLE. Bouncycastle. https://www.bouncycastle.org/.

[5] BÜNZ, B., BOOTLE, J., BONEH, D., POELSTRA, A., WUILLE, P., AND MAXWELL, G. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)* (2018), pp. 315–334.

[6] CHASE, M., AND LYSYANSKAYA, A. On signatures of knowledge. In *Advances in Cryptology - CRYPTO 2006* (Berlin, Heidelberg, 2006), C. Dwork, Ed., Springer Berlin Heidelberg, pp. 78–96.

[7] CHRIST, M., AND BONNEAU, J. Limits on revocable proof systems, with implications for stateless blockchains. In *Financial Cryptography and Data Security* (Cham, 2024), F. Baldimtsi and C. Cachin, Eds., Springer Nature Switzerland, pp. 54–71.

[8] DOBSON, S., GALBRAITH, S., AND SMITH, B. Trustless unknown-order groups. *Mathematical Cryptology 1*, 2 (Mar. 2022), 25–39.

[9] EAGEN, L., KANJALKAR, S., RUFFING, T., AND NICK, J. Bulletproofs++: Next generation confidential transactions via reciprocal set membership arguments. In *Advances in Cryptology – EUROCRYPT 2024: 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26–30, 2024, Proceedings, Part V* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 249–279.

[10] ESGIN, M. F., STEINFELD, R., AND ZHAO, R. K. Matrict+: More efficient post-quantum private blockchain payments. In *2022 IEEE Symposium on Security and Privacy (SP)* (2022), pp. 1281–1298.

[11] ESGIN, M. F., ZHAO, R. K., STEINFELD, R., LIU, J. K., AND LIU, D. Matrict: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2019), CCS '19, Association for Computing Machinery, p. 567–584.

[12] FIAT, A., AND SHAMIR, A. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO' 86* (1987).

[13] HABÖCK, U. Multivariate lookups based on logarithmic derivatives. Cryptology ePrint Archive, Paper 2022/1530, 2022. https://eprint.iacr.org/2022/1530.

[14] KURT, K., ALONSO, M., AND SARANG, N. Zero to monero - second edition, a technical guide to a private digital currency; for beginners, amateurs, and experts. https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf, 2020.

[15] LAI, R. W. F., RONGE, V., RUFFING, T., SCHRÖDER, D., THYAGARAJAN, S. A. K., AND WANG, J. Omniring: Scaling private payments without trusted setup. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019), CCS '19, Association for Computing Machinery, p. 31–48.

[16] MONERO. How much are monero transaction fees? https://www.monero.how/monero-transaction-fees, 2024.

[17] NAN, W., AND DONGXI, L. Flashswift: A configurable and more efficient range proof with transparent setup. In *Proceedings on Privacy Enhancing Technologies* (07 2024), vol. 2024.

[18] PIPPENGER, N. On the evaluation of powers and monomials. *SIAM J. Comput. 9*, 2 (may 1980), 230–250.

[19] POMPOSI, S. Tps or ttf? understanding blockchain speed. https://blog.fantom.foundation/tps-or-ttf-understanding-blockchain-speed/, 2021.

[20] SHEN NOETHER, A. M., AND TEAM, M. C. Ring confidential transactions. https://web.getmonero.org/resources/research-lab/pubs/MRL-0005.pdf, 2016.

[21] SRINIVASAN, S., KARANTAIDOU, I., BALDIMTSI, F., AND PAPAMANTHOU, C. Batching, aggregation, and zero-knowledge proofs in bilinear accumulators. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2022), CCS '22, Association for Computing Machinery, p. 2719–2733.

[22] WANG, N., AND CHAU, S. C.-K. Flashproofs: Efficient zero-knowledge arguments of range and polynomial evaluation with transparent setup. In *Advances in Cryptology – ASIACRYPT 2022* (2022), S. Agrawal and D. Lin, Eds., pp. 219–248.

[23] WANG, N., CHAU, S. C.-K., AND LIU, D. Swiftrange: A short and efficient zero-knowledge range argument for confidential transactions and more. In *2024 IEEE Symposium on Security and Privacy (SP)* (Los Alamitos, CA, USA, may 2024), IEEE Computer Society, pp. 54–54.

[24] YUEN, T. H., SUN, S.-F., LIU, J. K., AU, M. H., ESGIN, M. F., ZHANG, Q., AND GU, D. Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security. In *Financial Cryptography and Data Security* (Cham, 2020), J. Bonneau and N. Heninger, Eds., Springer International Publishing, pp. 464–483.

[25] ZHENG, T., GAO, S., SONG, Y., AND XIAO, B. Leaking arbitrarily many secrets: Any-out-of-many proofs and applications to ringct protocols. In *2023 IEEE Symposium on Security and Privacy (SP)* (Los Alamitos, CA, USA, may 2023), IEEE Computer Society, pp. 2533–2550.

A Inner Product Protocol Π_{IP}

The inner product protocol Π_{IP} is defined with the inputs $(\mathbf{g}, \mathbf{h}, Q, \mathbf{b}, \mathbf{a})$. The recursive composition of the protocol Π_{IP} is as below for $|\mathbf{b}| = |\mathbf{a}| > 1$:

If $|\mathbf{a}| = |\mathbf{b}| = 1$:

$$\mathcal{P} \Rightarrow \mathcal{V} : \mathbf{a},\ \mathbf{b}$$

$$\mathcal{V} : \mathbf{g}^{\mathbf{b}} \cdot \mathbf{h}^{\mathbf{a}} \cdot \tau^{\langle \mathbf{b}, \mathbf{a} \rangle} \stackrel{?}{=} Q$$

Else :

$$\mathcal{P} \Rightarrow \mathcal{V} : t_L = \langle \mathbf{b}_R, \mathbf{a}_L \rangle,\ t_R = \langle \mathbf{b}_L, \mathbf{a}_R \rangle$$

$$L = \mathbf{g}_L^{\mathbf{b}_R} \cdot \mathbf{h}_R^{\mathbf{a}_R} \cdot \tau^{t_L},\ R = \mathbf{g}_R^{\mathbf{b}_L} \cdot \mathbf{h}_L^{\mathbf{a}_R} \cdot \tau^{t_R}$$

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

$$\mathcal{P} \text{ and } \mathcal{V} : \mathbf{g}' = \mathbf{g}_L^{c} \circ \mathbf{g}_R^{c^{-1}},\ \mathbf{h}' = \mathbf{h}_L^{c^{-1}} \circ \mathbf{h}_R^{c}$$

$$Q' \triangleq L^{c^2} \cdot P \cdot R^{c^{-2}}$$

$$\mathcal{P} : \mathbf{b}' = c\mathbf{b}_R + c^{-1}\mathbf{b}_L,\ \mathbf{a}' = c^{-1}\mathbf{a}_R + c\mathbf{a}_L$$

Recursively run Π_{IP} on input $(\mathbf{g}', \mathbf{h}', Q', \mathbf{b}', \mathbf{a}')$

B Cryptographic Primitives

We adhere to the definitions presented in [22] to formalize homomorphic commitment schemes and zero-knowledge arguments of knowledge in the following sections.

B.1 Homomorphic Commitment Scheme

The scheme comprises two probabilistic polynomial-time (PPT) algorithms, denoted as $\mathcal{G}$ and Cm. The setup algorithm $\mathcal{G}(\lambda)$ generates a commitment key denoted as ck, while the commitment algorithm Cm defines a function $\mathsf{Cm}_{\mathsf{ck}} : \mathsf{M}_{\mathsf{ck}} \times \mathsf{R}_{\mathsf{ck}} \rightarrow \mathsf{C}_{\mathsf{ck}}$. In this scheme, M_{ck} represents the message space, R_{ck} represents the randomness space, and C_{ck} represents the commitment space. For any given message $m \in \mathsf{M}_{\mathsf{ck}}$, a uniformly random value $r \in \mathsf{R}_{\mathsf{ck}}$ is selected to compute a commitment $\mathsf{Cm}_{\mathsf{ck}}(m; r)$.

Definition 4 (Hiding) *A commitment scheme* $(\mathcal{G}, \mathsf{Cm})$ *is hiding if a commitment does not reveal the value for all PPT adversaries* $\mathcal{A}$*:*

$$Pr\left[\begin{array}{l} C = \mathsf{Cm}_{\mathsf{ck}}(m_b), \\ b \in \{0,1\}, \\ b' \leftarrow \mathcal{A}(C),\ b = b' \end{array} \middle| \begin{array}{l} \mathsf{ck} \leftarrow \mathcal{G}(\lambda), \\ (m_0, m_1 \in \mathsf{M}_{\mathsf{ck}}) \leftarrow \mathcal{A}(\mathsf{ck}) \end{array}\right] \approx \frac{1}{2}$$

The scheme is perfectly hiding if the probability is $\frac{1}{2}$.

Definition 5 (Binding) *A commitment scheme* $(\mathcal{G}, \mathsf{Cm})$ *is binding if a commitment can only be opened to one value for all PPT adversaries* $\mathcal{A}$*:*

$$Pr\left[\begin{array}{l} \mathsf{Cm}_{\mathsf{ck}}(m_0; r_0) \\ = \mathsf{Cm}_{\mathsf{ck}}(m_1; r_1), \\ m_0 \neq m_1 \end{array} \middle| \begin{array}{l} \mathsf{ck} \leftarrow \mathcal{G}(\lambda), \\ (m_0, m_1) \in \mathsf{M}_{\mathsf{ck}}, \\ (r_0, r_1) \in \mathsf{R}_{\mathsf{ck}} \end{array} \leftarrow \mathcal{A}(\mathsf{ck})\right] \leq \mathsf{negl}(\lambda)$$

The scheme is perfectly binding if the probability is 0.

B.2 Zero-Knowledge Arguments of Knowledge

A zero-knowledge argument is comprised of three interactive probabilistic polynomial-time algorithms $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$, where the setup algorithm $\mathsf{Setup}(\lambda)$ returns a common reference string σ. $\mathcal{P}$ and $\mathcal{V}$ are the prover and verifier algorithms, which produce the public transcript, $tr \leftarrow \langle \mathcal{P}(v), \mathcal{V}(t) \rangle$ on inputs v and t. Denote a polynomial-time decidable tertiary relation by $\mathrm{R} \subset \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$. A CRS-dependent language can be defined as $L_\sigma = \{u \mid \exists \omega : (\sigma, u, \omega) \in \mathrm{R}\}$, where ω is a witness for a statement u in the relation $(\sigma, u, \omega) \in \mathrm{R}$.

Definition 6 (Argument of Knowledge) *The triple* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *is called an argument of knowledge for the relation* R *if it satisfies the perfect completeness and computational witness-extended emulation.*

Definition 7 (Perfect Completeness) *An argument of knowledge* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *has perfect completeness if for all PPT adversaries* $\mathcal{A}$*:*

$$Pr\left[\begin{array}{l} (\sigma, u, \omega) \notin \mathrm{R} \text{ or} \\ \langle \mathcal{P}(\sigma, u, \omega), \mathcal{V}(\sigma, u) \rangle = 1 \end{array} \middle| \begin{array}{l} \sigma \leftarrow \mathsf{Setup}(\lambda), \\ (u, \omega) \leftarrow \mathcal{A}(\sigma) \end{array}\right] = 1$$

Definition 8 (Public Coin) *An argument of knowledge (*Setup*, $\mathcal{P}$, $\mathcal{V}$) is called public coin if the verifier chooses her messages uniformly at random and independently of the messages sent by the prover.*

Definition 9 (Computational Witness-Extended Emulation) *An argument of knowledge (*Setup*, $\mathcal{P}$, $\mathcal{V}$) has witness-extended emulation if for all deterministic polynomial time $\mathcal{P}^*$, there exists an expected polynomial time emulator $\mathcal{E}$ such that for all PPT adversaries $\mathcal{A}$:*

$$Pr\left[\mathcal{A}(tr)=1 \;\middle|\; \begin{array}{l}\sigma \leftarrow \mathsf{Setup}(\lambda)\\ (u,s) \leftarrow \mathcal{A}(\sigma),\\ tr \leftarrow \mathcal{O}\end{array}\right] \approx$$
$$Pr\left[\begin{array}{l}\mathcal{A}(tr)=1\\ \wedge\, tr \text{ is accepting}\\ \rightarrow (\sigma,u,w)\in \mathrm{R}\end{array} \;\middle|\; \begin{array}{l}\sigma \leftarrow \mathsf{Setup}(\lambda),\\ (u,s) \leftarrow \mathcal{A}(\sigma),\\ (tr,\omega) \leftarrow \mathcal{E}^{\mathcal{O}}(\sigma,u)\end{array}\right]$$

where the oracle is defined as $\mathcal{O} = \langle \mathcal{P}^*(\sigma,u,s), \mathcal{V}(\sigma,u)\rangle$.

Definition 10 (Perfect SHVZK) *A public coin argument of knowledge (*Setup*, $\mathcal{P}$, $\mathcal{V}$) is called perfect special honest verifier zero-knowledge argument of knowledge (SHVZK) for* R *if there exists a PPT simulator $\mathcal{S}$ such that for all interactive PPT adversaries $\mathcal{A}$:*

$$Pr\left[\begin{array}{l}(\sigma,u,\omega)\in \mathrm{R}\\ \wedge\, \mathcal{A}(tr)=1\end{array} \;\middle|\; \begin{array}{l}\sigma \leftarrow \mathsf{Setup}(\lambda),\\ (u,\omega,e) \leftarrow \mathcal{A}(\sigma),\\ tr \leftarrow \langle \mathcal{P}(v), \mathcal{V}(t)\rangle\end{array}\right] =$$
$$Pr\left[\begin{array}{l}(\sigma,u,\omega)\in \mathrm{R}\\ \wedge\, \mathcal{A}(tr)=1\end{array} \;\middle|\; \begin{array}{l}\sigma \leftarrow \mathsf{Setup}(\lambda),\\ (u,\omega,e) \leftarrow \mathcal{A}(\sigma),\\ tr \leftarrow \mathcal{S}(u,e)\end{array}\right]$$

where e is a public coin challenge, $v = (\sigma,u,\omega)$ and $t = (\sigma,u,e)$.

C Security Proofs

C.1 A Useful Lemma

Inspired by the work [17], we leverage the forking lemma from the study [3] to help with the security proofs of our protocols. The forking lemma naturally extends the concept of special soundness to public-coin protocols with $(2\mu+1)$ moves. Leveraging this lemma, we will demonstrate that our protocols achieve witness-extended emulation.

Lemma 3 (Forking Lemma) *Let* Setup, $\mathcal{P}$, $\mathcal{V}$ *be a $(2\mu+1)$-move, public coin interactive protocol. Let $\mathcal{E}$ be a witness extraction algorithm that succeeds with probability $1-\mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$ in extracting a witness from an $(\gamma_1,\ldots,\gamma_\mu)$-tree of accepting transcripts in probabilistic polynomial time. Assume that $\prod_{i=1}^{\mu}\gamma_i$ is bounded above by a polynomial in the security parameter λ. Then* Setup, $\mathcal{P}$, $\mathcal{V}$ *has witness-extended emulation.*

C.2 Theorem 1

Proof. **Perfect completeness** follows by carefully inspecting the equations in Eqn. (24) and Eqn. (25) for all valid witnesses.

Then we describe a **perfect SHVZK** simulation. Given all random challenges and a public-key ring $(P_i)_{i=1}^{|\mathcal{R}|}$, a simulator can simulate by randomly choosing group elements (T_2, C), two field vectors $(\mathbf{u}, \mathbf{n})$, and three field elements $(w, \hat{t}, r_{\hat{t}})$ so that T_1 and B can be uniquely determined according to the following equations:

$$T_1 = (g^{\hat{t}-z^2\cdot|\mathcal{S}|-\delta(y,z)} \cdot h^{r_{\hat{t}}} \cdot T_2^{-e^2})^{e^{-1}}$$
$$B = \prod_{i=1}^{|\mathcal{R}|}(g_i \cdot P_i^{dy^i})^{u_i} \cdot (h_i^{y^{-i}})^{n_i} \cdot h^{r_q}$$
$$\cdot\, \tau^{-wd} \cdot C^{-e} \cdot \prod_{i=1}^{|\mathcal{R}|}(g_i \cdot P_i^{dy^i})^{z} \cdot h_i^{-z}$$

By the perfectly hiding property, the Pedersen commitments in a real argument are uniformly random, as in the simulation. The field elements in a real argument are also uniformly random due to the prover's random choices of $(r_{u_i})_{i=1}^{|\mathcal{R}|}$, $(r_{n_i})_{i=1}^{|\mathcal{R}|}$, $r_{t_1}, r_{t_2}, r_w, r_c$ and r_b. Therefore, we have identical distributions of real and simulated arguments for the given challenges.

Finally, we prove **computational** $(|\mathcal{S}|+1)$**-special soundness**. Given the verification equation in Eqn. (25), in terms of the generators $(g_i)_{i=1}^{|\mathcal{R}|}$ and $(h_i)_{i=1}^{|\mathcal{R}|}$, where the DLOG-relations among them are non-trivial, we can determine the i-th elements of the two vectors $\mathbf{u}$ and $\mathbf{n}$ satisfy the linear forms $u_i \triangleq b_i - z + r_{u_i}\cdot e$ and $n_i \triangleq y^i \cdot (a_i + z + r_{n_i}\cdot e)$, respectively, unless the prover breaks the binding property of Pedersen commitment scheme. For each $i \in \{1,\ldots,|\mathcal{R}|\}$, an emulator interacts with the prover using random challenges and rewinds the prover to obtain two accepting transcripts. From these transcripts, the witnesses b_i and a_i can be easily extracted. By applying Bulletproofs' techniques and according to Eqn. (24), $\langle \mathbf{u},\mathbf{n}\rangle = z^2|\mathcal{S}| + \delta(y,z) + t_1\cdot e + t_2\cdot e^2$, it follows that $b_i \in \{0,1\}$, $\forall i \in \{1,\ldots,|\mathcal{R}|\}$.

Next, we try to extract a valid w. Given the challenge d, we can obtain the equality with respect to the generator τ alone by integrating the vector $\mathbf{u} = (u_i \triangleq b_i - z + r_{u_i}\cdot e)_{i=1}^{|\mathcal{R}|}$ into Eqn. (25):

$$(\tau^w)^d = \Big(\prod_{i=1}^{|\mathcal{R}|}(P_i^{y^i})^{u_i+z}\Big)^d \cdot \underbrace{\Big(\big(\prod_{i=1}^{|\mathcal{R}|}P_i^{y^i r_{u_i}}\big)^d \cdot \tau^{-r_w}\Big)^{-e}}_{\text{in } C} \tag{55}$$

Note that $(\prod_{i=1}^{|\mathcal{R}|} P_i^{y^i r_{u_i}})^d \cdot \tau^{-r_w}$ must be hidden in the commitment C, which is provided after the prover seeing the challenge d. Thus, by the binding property, we have:

$$\tau^w = \prod_{k=1}^{|\mathcal{S}|} P_k^{y^{\phi(k)}} \cdot \tau^{r_w \cdot e} \Longrightarrow w = \sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)} + r_w \cdot e \quad (56)$$

Finally, the emulator runs the argument with random challenges and rewinds the prover until it obtains $(|\mathcal{S}|+1)$ accepting transcripts. We compute a challenge matrix $\mathbf{y}$, which are invertible since all the rows and columns are linearly independent:

$$\boldsymbol{y} = \begin{pmatrix} y_1^{\phi(1)} & \dots & y_1^{\phi(|\mathcal{S}|)} & e_1 \\ \vdots & \ddots & \vdots & \vdots \\ y_{|\mathcal{S}|+1}^{\phi(1)} & \dots & y_{|\mathcal{S}|+1}^{\phi(|\mathcal{S}|)} & e_{|\mathcal{S}|+1} \end{pmatrix}$$

We can obtain the secret witnesses $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$ and the random value r_w by computing:

$$\begin{pmatrix} s_{\phi(1)} \\ \vdots \\ s_{\phi(|\mathcal{S}|)} \\ r_w \end{pmatrix} = \boldsymbol{y}^{-1} \cdot \begin{pmatrix} w_1 \\ \vdots \\ w_{|\mathcal{S}|+1} \end{pmatrix}$$

Once we have valid $\mathbf{u}$, $\mathbf{n}$ and w, we can integrate them into equations to obtain the openings of the involved commitments.

C.3 Theorem 2

Proof. **Perfect completeness** follows by carefully inspecting the equations in Eqn. (43) and Eqn. (44) for all valid witnesses. Please see Section 7.3.2 for full details.

Then we describe a **perfect SHVZK** simulation. Given all random challenges and a set of tags $(T_k)_{k=1}^{|\mathcal{S}|}$, a simulator can simulate the transcript by randomly choosing group elements (M_2, Z, E), three field vectors $(\mathbf{u}, \mathbf{v}, \mathbf{f})$, and three field elements $(w, \hat{m}, r_{\hat{m}})$ so that M_1 and Y can be uniquely determined according to the equations:

$$\begin{aligned} M_1 &= (g^{\hat{m} - \sum_{k=1}^{|\mathcal{S}|} c^k} \cdot h^{r_{\hat{m}}} \cdot M_2^{-e^2})^{e^{-1}} \\ Y &= \prod_{i=1}^{|\mathcal{R}|} (\epsilon^{-d^3 (y^i + z)^{-1}})^{u_i} \cdot \prod_{k=1}^{|\mathcal{S}|} \left((g_k \cdot T_k^{d^2})^{c^{-k}}\right)^{v_k} \\ &\quad \cdot \prod_{k=1}^{|\mathcal{S}|} (h_k^d \cdot \epsilon^{d^3})^{f_k} \cdot h^{r_s} \cdot \eta^{-wd^2} \cdot \prod_{k=1}^{|\mathcal{S}|} T_k^{-d^2 z} \\ &\quad \cdot \epsilon^{-d^3 z \sum_{i=1}^{|\mathcal{R}|} (y^i + z)^{-1}} \cdot (\prod_{k=1}^{|\mathcal{S}|} g_k)^{-z} \cdot Z^{-d} \cdot E^{-e} \end{aligned}$$

By the perfectly hiding property, the Pedersen commitments in a real argument are uniformly random, as in the simulation. The field elements in a real argument are also uniformly random due to the prover's random choices of $(r_{v_k})_{k=1}^{|\mathcal{S}|}$, $(r_{f_k})_{k=1}^{|\mathcal{S}|}$, r_{m_1}, r_{m_2}, r_y, r_e and r_z. Therefore, we have identical distributions of real and simulated arguments for the given challenges.

Finally, we prove **computational soundness**. Note that the soundness builds upon the validity of the vector $\mathbf{u}$ in K-out-of-N proof. First, in terms of the two generator vectors $(g_k)_{k=1}^{|\mathcal{S}|}$ and $(h_k)_{k=1}^{|\mathcal{S}|}$ from the verification equation (44), we can determine that v_k and f_k for $k \in \{1, ..., |\mathcal{S}|\}$ satisfy the following linear forms, respectively:

$$\begin{aligned} v_k &= c^k \cdot (v_k^{(0)} + z + v_k^{(1)} \cdot d + v_k^{(2)} \cdot d^2 z + v_k^{(3)} \cdot e) \\ f_k &= f_k^{(0)} + f_k^{(1)} \cdot dz + f_k^{(2)} \cdot e \end{aligned}$$

where we use a tuple of variables $(v_k^{(0)}, v_k^{(1)}, v_k^{(2)}, v_k^{(3)}, f_k^{(0)}, f_k^{(1)}, f_k^{(2)})$ in the equations as their values have not been determined yet:

- $(v_k^{(0)})_{k=1}^{|\mathcal{S}|}$ refers to the witness vector hidden in Y.
- $(v_k^{(1)})_{k=1}^{|\mathcal{S}|}$ and $(f_k^{(0)})_{k=1}^{|\mathcal{S}|}$ refer to the witness vectors hidden in Z.
- $(v_k^{(2)})_{k=1}^{|\mathcal{S}|}$ and $(f_k^{(1)})_{k=1}^{|\mathcal{S}|}$ refer to the witness vectors hidden in $(T_k)_{k=1}^{|\mathcal{S}|}$.
- $(f_k^{(2)})_{k=1}^{|\mathcal{S}|}$ refers to the witness vector hidden in E.

Note that in terms of the challenge d, the actual f_k should have taken the form $f_k^{(0)} + f_k^{(1)} \cdot dz + f_k^{(2)} \cdot \frac{e}{d}$ based on the right-hand side of Eqn. (44). However, given the valid form of v_k, computing $v_k \cdot f_k$ in Eqn. (43) does not yield terms with exponents $\frac{e^2}{d}$. Since the commitment E is revealed only after the challenge d is observed, embedding d in E ensures that f_k is restricted to include only the term $f_k^{(2)} \cdot e$. By the binding property of Eqn. (43), where $\hat{m} = \sum_{k=1}^{|\mathcal{S}|} c^k + m_1 \cdot e + m_2 \cdot e^2$, we have the following equality for $k \in \{1, ..., |\mathcal{S}|\}$:

$$c^k \cdot (v_k^{(0)} + z + v_k^{(1)} \cdot d + v_k^{(2)} \cdot d^2 z) \cdot (f_k^{(0)} + f_k^{(1)} \cdot dz) = c^k$$

We can safely cancel out the non-zero factor c^k on both sides:

$$(v_k^{(0)} + z + v_k^{(1)} \cdot d + v_k^{(2)} \cdot d^2 z) \cdot (f_k^{(0)} + f_k^{(1)} \cdot dz) = 1$$

We can then rewrite the left-hand side of the equality as a polynomial in terms of the challenge d, as shown below:

$$\begin{aligned} &(v_k^{(0)} + z + v_k^{(1)} \cdot d + v_k^{(2)} \cdot d^2 z) \cdot (f_k^{(0)} + f_k^{(1)} \cdot dz) = 1 \\ \Rightarrow\ &\underbrace{(v_k^{(0)} + z) \cdot f_k^{(0)}}_{=1} + \underbrace{(v_k^{(0)} + z) \cdot f_k^{(1)}}_{=0} \cdot dz + \underbrace{v_k^{(1)} \cdot f_k^{(0)}}_{=0} \cdot d \\ &+ \underbrace{(v_k^{(1)} \cdot f_k^{(1)} + v_k^{(2)} \cdot f_k^{(0)})}_{=0} \cdot d^2 z + \underbrace{v_k^{(2)} \cdot f_k^{(1)}}_{=0} \cdot d^3 z^2 = 1 \end{aligned}$$

By the Schwartz-Zippel lemma, the above equation yields 5 constraints. Specifically, the first constraint $(v_k^{(0)}+z)\cdot f_k^{(0)}=1$ gives us $(v_k^{(0)}+z)\neq 0 \wedge f_k^{(0)}\neq 0$ and the reciprocal relation $f_k^{(0)}=(v_k^{(0)}+z)^{-1}$. Then we can learn that $v_k^{(1)}=v_k^{(2)}=f_k^{(1)}=0$ from the other four constraints with overwhelming probability. Thus, we have the following linear forms satisfying Eqn. (39) and Eqn. (40), respectively:

$$v_k = c^k\cdot(v_k^{(0)}+z+v_k^{(3)}\cdot e)$$
$$f_k = f_k^{(0)}+f_k^{(2)}\cdot e$$

An emulator rewinds the prover to obtain two accepting transcripts to extract $(v_k^{(0)}, v_k^{(3)})$ and $(f_k^{(0)}, f_k^{(2)})$, respectively.

Next, we try to extract the permutation relation between $(v_k^{(0)})_{k=1}^{|\mathcal{S}|}$ and $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$. With respect to the generator ϵ^{d^3} alone, by integrating the values $(u_i=b_i-z+r_{u_i}\cdot e)_{i=1}^{|\mathcal{R}|}$, we can conclude:

$$\sum_{k=1}^{|\mathcal{S}|} f_k^{(0)} = \sum_{i=1}^{|\mathcal{R}|} b_i\cdot(y^i+z)^{-1} = \sum_{k=1}^{|\mathcal{S}|}(y^{\phi(k)}+z)^{-1}$$

Given $f_k^{(0)}=(v_k^{(0)}+z)^{-1}$, we have the following equality:

$$\sum_{k=1}^{|\mathcal{S}|}(v_k^{(0)}+z)^{-1} = \sum_{k=1}^{|\mathcal{S}|}(y^{\phi(k)}+z)^{-1}$$

Recall that the commitment Y concealing $(v_k^{(0)})_{i=1}^{|\mathcal{S}|}$ is provided before seeing the random challenge z. Due to the uniformly randomness of z in the large space $\mathbb{Z}_p^*$, we have the fact that $(v_k^{(0)})_{k=1}^{|\mathcal{S}|}$ is a permutation of the sequence $(y^{\phi(k)})_{k=1}^{|\mathcal{S}|}$ unless the PPT prover is able to break the binding property of the commitment scheme.

By integrating the value $w=\sum_{k=1}^{|\mathcal{S}|} y^{\phi(k)} s_{\phi(k)}+r_w\cdot e$ from our K-out-of-N proof into Eqn. (52) and by the Schwartz-Zippel lemma, it is with overwhelming probability that the secret keys hidden in the tags $(T_k)_{k=1}^{|\mathcal{S}|}$ form a permutation of the secret keys $(s_{\phi(k)})_{k=1}^{|\mathcal{S}|}$ in the source public keys. Finally, once we have valid $\mathbf{v}$, $\mathbf{f}$ and w, we can integrate them into equations to obtain the openings of the involved commitments.

C.4 Tag Properties

Pseudo-randomness. The pseudo-randomness of tags is implied by the well-known Decisional Diffie-Hellman (DDH) assumption. The DDH assumption states that given two group elements $g^a\in\mathbb{G}$ and $g^b\in\mathbb{G}$ for uniformly and independently chosen $a,b\in\mathbb{Z}_p^*$, $g^{ab}\in\mathbb{G}$ looks like a random element in $\mathbb{G}$. Thus, given two randomly chosen generators $\tau,\eta\xleftarrow{\$}\mathbb{G}$ such that $\eta=\tau^u$ and a public key $P=\tau^s$, where $s,u\xleftarrow{\$}\mathbb{Z}_p^*$ are secret and randomly chosen, it is computationally infeasible for a PPT adversary to distinguish the tag $T=\eta^s=\tau^{us}$ from a random element in $\mathbb{G}$ under the DDH assumption.

One-wayness. We can prove it by the DLOG assumption. Given a secret key $s\in\mathbb{Z}_p^*$, it is easy to compute $T=\eta^s$, but computationally hard to obtain s from T.