

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Just One Bit Vector: Complement-Free Ring Confidential Transactions with Transparent Setup

Hao Gao and Qianhong Wu, *School of Cyber Science and Technology, Beihang University; Hangzhou Innovation Institute of Beihang University;* Bo Qin, *Renmin University of China;* Fudong Wu, *School of Cyber Science and Technology, Beihang University; Hangzhou Innovation Institute of Beihang University;* Zhenyang Ding, *School of Cyber Science and Technology, Beihang University*

https://www.usenix.org/conference/usenixsecurity26/presentation/gao-hao

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

Just One Bit Vector: Complement-Free Ring Confidential Transactions with Transparent Setup

Hao Gao[*,†] Qianhong Wu[*,†] Bo Qin[‡,✉] Fudong Wu[*,†] Zhenyang Ding[*]

Abstract

Ring Confidential Transactions (RingCTs) rely on their signature schemes to ensure transaction validity while preserving user privacy. In the discrete-logarithm setting, state-of-the-art RingCT signatures leverage Bulletproof as their backbone. In this work, we conduct a thorough analysis of existing Bulletproof-based RingCT signatures and identify two pervasive sources of inefficiency: (i) redundant complement advice induced by binary constraints, and (ii) substantial overhead incurred by defenses against the Knowledge-of-Known-Discrete-Logarithm (KKDL) issue. We present new techniques that eliminate the first inefficiency and significantly reduce the second. We introduce the Weighted-Norm Inner Linear Argument (WNILA) as a new compression pivot and leverage it to design zero-knowledge proofs that capture constraints arising in RingCT without introducing complement advice. We then adopt the techniques of Omniring and Lin et al. to unify the K-Weight and K-Link techniques proposed by ZGSX23 and BulletCT. Building on this framework, we derive three RingCT signature schemes: UniBit, UniBit^{+}, and UniBitRange. We implement all three schemes and benchmark them against BulletCT and Omniring* under practical parameter regimes. Our experimental results demonstrate 2.5–40$\times$ improvements in signing and verification speed over Omniring*, and 1.6–2.3$\times$ signing speedups together with up to 1.65$\times$ verification speedup over BulletCT, while achieving comparable signature sizes.

1 Introduction

Blockchains have become a financial infrastructure underlying cryptocurrencies, with billions of dollars' worth of assets traded every day. The widespread adoption of blockchains has raised significant concerns about user privacy. In most deployed blockchain systems, transaction data are recorded in plaintext on publicly accessible ledgers. As a result, on-chain activity can often be linked to real-world identities [2, 6, 19]. These concerns have motivated research on privacy-enhancing technologies. One prominent solution is Ring Confidential Transactions (RingCT) [25], a cryptographic protocol that enables private payments on UTXO-based blockchains. RingCT forms the cryptographic core of Monero [24], one of the most popular privacy-centric cryptocurrencies.

RingCT employs multiple cryptographic techniques to achieve transaction privacy. The central challenge is to ensure the validity of a spending transaction without revealing any sensitive information. In particular, a RingCT transaction must hide both the transaction amounts and the identities of the input owners. To hide transaction amounts, RingCT employs cryptographic commitments to conceal the transferred amounts. To hide input identities, RingCT mixes each real input with a ring of decoy inputs. This design makes transaction validity checking inherently challenging, since verification must proceed without knowing which inputs are real.

Monero employs the CLSAG [14] linkable ring signature (LRS) scheme to hide input ownership while enabling public verification and linkability. Because the size of a CLSAG signature grows linearly with both the ring size and the number of real inputs, Monero fixes its ring size at 16 to balance anonymity and efficiency. Such a small ring may expose the system to de-anonymization risks [1, 17, 29], motivating the development of RingCT signature schemes that scale efficiently to much larger rings. Although increasing the ring size also enlarges the transaction size, techniques such as [10] may be used to reduce the ring description to logarithmic size.

ZKP-based RingCT signatures. To make RingCT scalable to larger rings, a line of work [18, 20, 22, 28, 30, 31] incorporates zero-knowledge proofs (ZKPs) into RingCT. Informally, the validity of a RingCT transaction with K inputs and M outputs can be captured by the following constraints:

- **K-out-of-N**, requiring that the spender knows at least K secret keys among N decoy public keys, where N

[*]School of Cyber Science and Technology, Beihang University, Beijing, China.
[†]Hangzhou Innovation Institute of Beihang University, Hangzhou, China.
[‡]Renmin University of China, Beijing, China.
[✉]Corresponding author, bo.qin@ruc.edu.cn

determines the anonymity level.

- **Linkability**, ensuring that a set of K transaction tags are correctly bound to the spender's hidden secret keys, allowing double-spending to be detected.
- **Range**, enforcing that output values hidden inside coin commitments lie within a valid range. This is crucial to prevent the "overflow" attack.
- **Balance**, enforcing conservation of value between the K inputs and the M outputs, where all values are hidden in cryptographic commitments.

These works construct a public-coin ZKP for the required RingCT constraints and then compile it into a *signature of knowledge* (SoK) [9] via the Fiat-Shamir heuristic [13]. The advantage of this paradigm is that it can leverage the succinctness of modern ZKP techniques to obtain short signatures. We note that Monero currently does not use any ZKP-based RingCT signature.

1.1 Bulletproof-based RingCT Signatures

In the discrete-logarithm (DL) setting, state-of-the-art RingCT signature schemes adopt the Bulletproof [8] technique as their backbone. Bulletproof is particularly well suited to RingCT, as it admits a transparent setup, achieves compact signatures, and integrates naturally with DL-based commitment schemes. Despite these advantages, existing Bulletproof-based designs leave substantial room for improvement. After a thorough analysis, we identify two pervasive sources of inefficiency in Bulletproof-based RingCT signature schemes that were not explicitly addressed in prior work: (i) redundant advice introduced by binary proofs, and (ii) the computational cost incurred by defenses against the Knowledge-of-Known-Discrete-Logarithm (KKDL) issue [22]. In what follows, we present a fine-grained analysis of these issues and their implications for the efficiency of RingCT signature schemes.

Overhead from the complement advice. To better understand the inefficiencies shared by Bulletproof-based RingCT signatures, we briefly revisit their common design template. Bulletproof-based RingCT signatures typically follow a three-stage workflow. First, an "embedding stage" incorporates public inputs into a commitment to a bit vector $\mathbf{n}$. Second, a linear-communication ZKP proves that the committed vector encodes a valid bit vector. Third, a "compression stage" applies Bulletproof's inner-product argument (IPA) to compress the linear-communication proof from the previous stage, yielding a succinct signature.

For example, both RingCT-3.0 [30] and Omniring [20] embed public inputs into a Pedersen vector commitment [26] over a bit vector of size $O(\mathrm{NK})$, and use a linear-communication Bulletproof-style ZKP that is subsequently compressed to logarithmic size via the IPA. Subsequent work

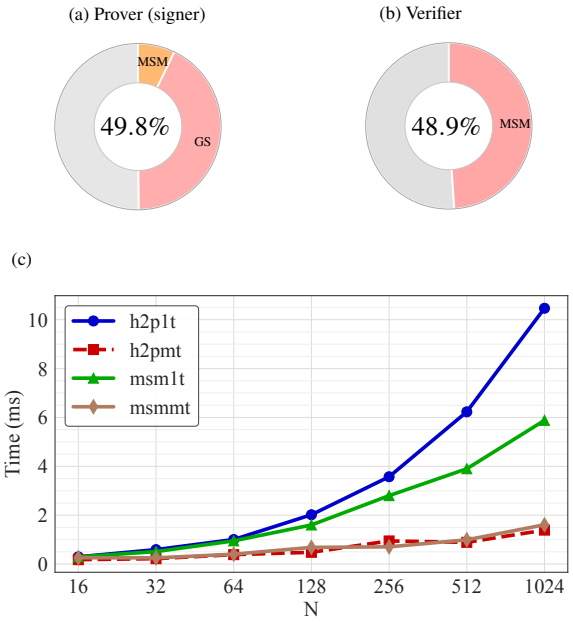

Figure 1: **(a), (b):** group-operation overhead caused by the complement advice for $\mathrm{N} = 1024$. MSM denotes multi-scalar multiplication and GS denotes single group-scalar multiplications. **(c):** hash-to-point versus same-size MSM runtime. 1t means single-threaded and mt means multi-threaded. We use 10 threads for mt and the benchmark is conducted on a MacBook M1 Pro.

by ZGSX23 [31] introduces the "K-Weight" technique to improve efficiency in the Any-out-of-N setting, while BulletCT [28] further augments this approach with the "K-Link" technique to support standard K-out-of-N RingCT signatures.

Across all Bulletproof-based RingCT signature schemes, the bit vector $\mathbf{n}$ encodes the positions of the real inputs within the decoy rings and hence has length at least N. These constructions enforce binary constraints using the standard IPA-based approach, which requires the signer to *additionally* commit to a "complement advice" vector $\mathbf{1} - \mathbf{n}$ and then prove $\mathbf{n} \circ (\mathbf{1} - \mathbf{n}) = \mathbf{0}$. This complement advice constitutes a major source of redundancy, as it effectively doubles the witness size for binary constraints.

On the signer side, the IPA proceeds via a round-by-round "self-folding" procedure, where the per-round cost scales linearly with the witness length. As a result, the complement advice directly translates into additional overhead for the signer's dominant group operations. On the verifier side, the final multi-scalar multiplication (MSM) involves at least $|\mathbf{n}|$ additional terms due to the complement advice.

Precisely quantifying the overhead introduced by complement advice in a general RingCT signature is challenging. Nevertheless, a meaningful estimate can be obtained by analyzing the structure of the IPA-based binary proof, which appears as a common subroutine in RingCT signatures. An IPA-

based binary proof essentially proves $\langle \mathbf{n} \circ \boldsymbol{\kappa}^{|\mathbf{n}|}, \mathbf{1} - \mathbf{n} \rangle = 0$ for a commitment of the form $\mathbf{G}^{\mathbf{n}}\mathbf{A}^{\mathbf{1}-\mathbf{n}}$. $\boldsymbol{\kappa}^{|\mathbf{n}|} = (1, \kappa, \ldots, \kappa^{|\mathbf{n}|-1})$ serves as the compression coefficients for the challenge κ. For every round, this IPA employs a self-folding technique to reduce the problem size by a factor of two. This particular structure makes a precise analysis of the overhead from the complement advice straightforward.

On the signer side, we estimate the overhead by summing the group operations incurred in each round of the IPA's self-folding procedure and then attributing half of the total cost to the complement advice. On the verifier side, we estimate the overhead by measuring the time spent in the final MSM and similarly attributing half of the total cost to the complement advice. Figure 1(a,b) reports the resulting overheads for the signer and verifier, respectively, for a standard IPA-based binary proof instantiated over Ristretto255. The results show that approximately half of the prover and verifier time spent on the binary proof can be attributed to operations induced by the complement advice.

The price to pay for safety. Applying Bulletproof to RingCT introduces a subtle but fundamental challenge. Bulletproof assumes that the generators used in the protocol do not admit non-trivial DL relations known to the prover. In RingCT, however, public inputs may be adversarially generated, giving rise to the Knowledge-of-Known-Discrete-Logarithm (KKDL) issue [22]. If protocol generators are derived from such adversarial inputs in an unsafe manner, the security guarantees of Bulletproof may no longer apply.

Lin et al. [22] analyze several prior RingCT signature schemes and observe that the KKDL issue is often not explicitly addressed in their security proofs. They present a concrete attack against RingCT-3.0 [30] and identify a subtle flaw in Omniring [20]. Balla and Chaidos [5] further demonstrate that similar gaps arise in several linkable ring signature schemes. In particular, they present attacks that break linkability against LLRings [18] and the threshold variant of RingReferral [27]. Across these constructions, a common root cause emerges: existing analyses do not explicitly account for the possibility that protocol generators may embed DL relations known to the prover. The KKDL issue is not specific to Bulletproof. Rather, it reflects a general challenge in applying proof systems whose security relies on concealing non-trivial DL relations from the adversary. For example, the pairing-based variant of LLRing [18] adopts Dory [21]. Dory similarly assumes that non-trivial DL relations are hard to compute.

Omniring [20] and Lin et al. [22] propose a generic technique known as "KKDL-Embedding" to address the KKDL issue. While effective, KKDL-Embedding inevitably incurs additional overhead. In practice, this cost is non-trivial and dominated by two main components:

- **Hash-to-point:** a number of cryptographically secure hash-to-point evaluations that scale linearly with the underlying Bulletproof witness size and are required by both the signer and the verifier;
- **Auxiliary protocol:** a compressed Σ-protocol [4] instantiated over an instance whose size scales linearly with the ring size N.

Metric	Omniring* [22]	UniBit & UniBit$^+$	UniBitRange
Hash-to-point	2δ	$\max(N+2K+9, \beta \cdot M)$	$N+2K+9+\beta \cdot M$
Size of Σ-protocol	$N+3$	$2K+8$	$2K+8$

Table 1: Summary of hash-to-point usage and Σ protocol instance size. $\delta = \beta M + N(K+1) + 3K + 3$.

The "hash-to-point" overhead arises from technical aspects inherent to the KKDL-Embedding, which requires the verifier to supply fresh random generators for each protocol execution. In the non-interactive setting, these generators are derived from the transcript using cryptographically secure hash-to-point functions. Such operations are typically more expensive than standard cryptographic hashes, such as SHA-256.

For instance, when the scheme is instantiated over Ristretto255, hash-to-point operations rely on Elligator-style maps [7] and incur a high practical cost. Our benchmarks show that their cost exceeds that of an MSM of the same size in the single-threaded setting, and that the two become roughly comparable when multi-threading is enabled. Since efficient parallelism is not always available, this overhead remains significant. Figure 1(c) compares the runtime of hash-to-point operations and MSMs.

The "auxiliary protocol" overhead arises from the security refinement of Omniring proposed by Lin et al. [22], which introduces an additional compressed Σ-protocol. On the signer side, this component incurs additional group operations that scale linearly with the ring size N. Since N is typically the dominant scaling parameter in RingCT, this overhead is significant in practice. The verifier is less affected, as verification reduces to an $O(N)$ MSM that can be batched.

Research gap. Each real-world RingCT transaction requires generating and verifying a RingCT signature, making the efficiency of the signature a direct determinant of overall system scalability. Motivated by the analysis above, we ask the following research question: *Is there a more efficient design paradigm for Bulletproof-based RingCT signature schemes?*

1.2 Our Contributions

1.2.1 New Techniques

We develop novel techniques that eliminate the need for complement advice in Bulletproof-based RingCT signature schemes and significantly reduce the overhead caused by the

Table 2: A complexity comparison among the best-known RingCT signatures under the discrete-logarithm setting.

Protocols	**Round-by-round cost for compression**	**Verifier's MSM**
UniBit (K/N)	$(\frac{3}{2}\mathrm{N}'+\frac{5}{2}\mathrm{K}')\mathbf{M}+(\frac{1}{2}\mathrm{N}'+\mathrm{K}')\mathbf{G}+\mathbf{P}_{\mathrm{RP}}$	$(3(\mathrm{N}+\mathrm{K})+2\log(\mathrm{N})+\mathrm{M}+28)+\mathbf{V}_{\mathrm{RP}}$
UniBit^{+} (K/N)	$(\frac{3}{2}\mathrm{N}'+\frac{5}{2}\mathrm{K}')\mathbf{M}+(\frac{1}{2}\mathrm{N}'+\mathrm{K}')\mathbf{G}+\mathbf{P}_{\mathrm{RP}}$	$(3(\mathrm{N}+\mathrm{K})+2\log(\mathrm{N})+\mathrm{M}+26)+\mathbf{V}_{\mathrm{RP}}$
UniBitRange (K/N)	$(\frac{3}{2}(\mathrm{N}'+(\beta\mathrm{M})')+\frac{5}{2}\mathrm{K}')\mathbf{M}+(\frac{1}{2}(\mathrm{N}'+(\beta\mathrm{M})')+\mathrm{K}')\mathbf{G}$	$(3(\mathrm{N}+\mathrm{K})+2\cdot\log(\max(\mathrm{N},\beta\mathrm{M},\mathrm{K})))+28+(\beta+1)\mathrm{M}$
Omniring [20] (K/N)	$(2\delta')\,\mathbf{M}+(\delta')\,\mathbf{G}$	$(2\mathrm{NK}+2\mathrm{N}+6\mathrm{K}+2\lceil\log\delta\rceil+(2\beta+1)\mathrm{M})+6$
BulletCT [28] (K/N)	$2\cdot(\mathrm{N}+\mathrm{K}+\beta\mathrm{M})'\,\mathbf{M}+(\mathrm{N}+\mathrm{K}+\beta\mathrm{M})'\,\mathbf{G}$	$(4\mathrm{N}+3\mathrm{K}+(2\beta+1)\mathrm{M}+2\log(\max(\mathrm{N},\beta\mathrm{M},\mathrm{K}))+19)$
ZGSX23 [31] (A/N)	$2(\mathrm{N}+\beta\mathrm{M})'\,\mathbf{M}+(\mathrm{N}+\beta\mathrm{M})'\,\mathbf{G}$	$(6\mathrm{N}+(2\beta+1)\mathrm{M})+2\log(\max(\beta\mathrm{M},\mathrm{N}))+12$

"K/N" and "A/N" indicate the K-out-of-N and Any-out-of-N settings for the RingCT signature schemes. We assume $\mathrm{N},\mathrm{K},\mathrm{M},\beta$ are powers of two, where β is the range bit-length, and define $\delta=\beta\mathrm{M}+\mathrm{N}(\mathrm{K}+1)+3\mathrm{K}+3$. We denote MSMs and single-scalar multiplications by $\mathbf{M}$ and $\mathbf{G}$, respectively. The notation $(\cdot)'$ represents quantities after the i-th folding round (e.g., $\delta'=\delta/2^{i}$). $\mathbf{P}_{\mathrm{RP}}$ and $\mathbf{V}_{\mathrm{RP}}$ denote the cost of an additional aggregated range-proof over M commitments.

KKDL-Embedding technique. Our techniques are summarized as follows.

A new compression pivot. We introduce the Weighted-Norm Inner Linear Argument (WNILA), which can serve as a drop-in replacement for Bulletproof's IPA as the compression layer in RingCT signature schemes.

Complement-free ZKP. Leveraging WNILA, we design ZKPs that capture the constraints arising in RingCT while eliminating the need for auxiliary complement advice.

KKDL-secure integration. We integrate the KKDL-Embedding technique with both the "K-weight" technique of ZGSX23 [31] and the "K-Link" technique of BulletCT [28], resulting in a unified design that remains secure in the presence of the KKDL issue.

1.2.2 New RingCT Signature Schemes

Building on this flexible framework, we derive three concrete RingCT signature schemes: UniBit, UniBit^{+}, and UniBitRange. All three schemes share the same technical core while exploring different trade-offs within the design space. These schemes follow the same high-level structure. We present UniBit in the main body of the paper. UniBit+ employs a more sophisticated blinding technique that reduces the signature size by two group elements and two field elements; however, this scheme is more technically involved. UniBitRange integrates the range constraint directly into the main constraint system. This design sacrifices flexibility in the choice of range-proof in exchange for improved communication efficiency. We provide the main ideas of UniBit+ and UniBitRange in Appendix C and D and leave the formal descriptions in the full version of the paper.

We present the costs incurred by hash-to-point operations and compressed Σ-protocols in our schemes, as well as in Omniring* [20] (the security-patched version of Omniring introduced by Lin et al. [22]), in Table 1.

Implementation and evaluation. We implement UniBit, UniBit^{+}, UniBitRange, Omniring*, and BulletCT and evaluate all five schemes over practical parameter regimes. Compared to Omniring* [22], our schemes improve signing and verification speed by roughly 2.5–40$\times$ across parameter settings ranging from $(\mathrm{N}=32,\mathrm{K}=1,\mathrm{M}=2)$ to $(\mathrm{N}=4096,\mathrm{K}=16,\mathrm{M}=2)$ with comparable signature size. Compared to BulletCT, the signing/verification improvement depends on the construction and parameter regime; in our evaluated settings, the observed improvement is roughly 1.6–2.3$\times$ for signing, and 1.1–1.65$\times$ for verification.

Theoretical efficiency. Table 2 reports a fine-grained theoretical efficiency comparison for signing and verification against other state-of-the-art schemes. Signer performance is dominated by two qualitatively different operations: MSMs and single-scalar multiplications, where the latter are significantly

more expensive than the amortized per-scalar cost in an MSM. The dominant signing cost arises in the compression stage, which consists of a round-by-round self-folding procedure involving MSMs together with single-scalar multiplications. Our constructions follow the same structure. Accordingly, we report both the number of MSMs and single-scalar multiplications performed *per round*. For schemes that employ an independent range-proof, including UniBit and UniBit^{+}, this metric also accounts for the cost of the separate range-proof component. We use this metric in Table 2 to highlight the dominant computational costs.

This metric conservatively *underestimates* the practical advantage of our schemes. For example, our techniques eliminate blinding commitments for complement advice, and UniBit and UniBit^{+} are compatible with 16-radix range-proofs such as Bulletproof^{++} [12]. In contrast, schemes that embed a 2-radix range proof into the main constraint system do not admit such optimizations. As shown in Table 2, our schemes require the fewest single-scalar multiplications and MSMs with respect to the ring size N. Moreover, when additionally accounting for UniBitRange, which deeply integrates the range constraints into the protocol design, UniBitRange also requires the fewest single-scalar multiplications and MSMs with respect to $\beta \cdot \mathrm{M}$.

1.2.3 Additional Applications

Any-out-of-N RingCT signatures. Any-out-of-N RingCT signatures hide the number of real inputs among N public keys [31]. Our framework naturally extends to this setting and yields an efficient Any-out-of-N RingCT construction in the discrete-logarithm setting, which is technically simpler than the K-out-of-N setting. The construction follows directly from adapting our techniques to the approach of Section 6.2.3 of BulletCT [28].

Novel range-proof. By abstracting out the range-proof component embedded in UniBitRange, we obtain a new range-proof in the discrete-logarithm setting. Our construction improves prover and verifier efficiency over Bulletproof and Bulletproof^{+} while achieving comparable proof size. It can also be viewed as a simpler alternative to Bulletproof^{++} [12]. We note that, as a standalone range-proof scheme, our novel range-proof construction has *worse* prover time, verifier time, and proof size than Bulletproof^{++}. Its primary advantages instead lie in its implementation simplicity and its natural compatibility with UniBitRange.

1.3 Overview of Our Techniques

Figure 2 illustrates the overall structure of our schemes. Our key insight is that the auxiliary advice incurred by binary constraints is not inherent to the constraints themselves, but rather an artifact of the IPA. This observation motivates an alternative compression technique that aligns more closely

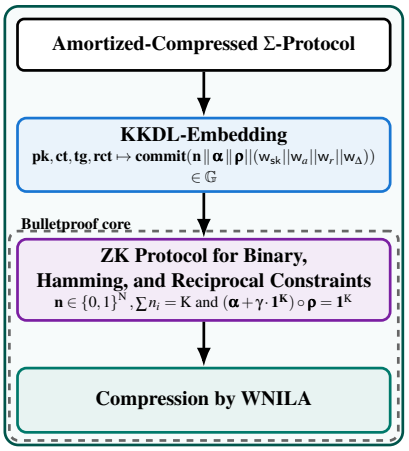

Figure 2: Structural diagram of our design

with the structure of binary constraints. WNILA is precisely such an alternative.

Concretely, given a committed vector $(\mathbf{n}, \mathbf{a}, \mathbf{b}, \mathbf{l}, v)$ and a public vector $\mathbf{c}$, WNILA enforces the relation $v = \|\mathbf{n}\|_{\mu}^2 + \langle \mathbf{a}, \mathbf{b} \rangle + \langle \mathbf{c}, \mathbf{l} \rangle$, where the weighted norm is defined as $\|\mathbf{n}\|_{\mu}^2 = \sum_{i=0}^{\mathrm{N}-1} \mu^{i+1} n_i^2$. The purpose of the weighted-norm term is to compress the binary constraint $\mathbf{n} \in \{0,1\}^{\mathrm{N}}$ via the random challenge μ as $\left\|\mathbf{n} - \frac{1}{2}\mathbf{1}^{\mathrm{N}}\right\|_{\mu}^2 = \sum_{i=0}^{\mathrm{N}-1} \mu^{i+1} \cdot \frac{1}{4}$. By expressing binary and Hamming-weight constraints directly as weighted-norm relations, WNILA eliminates the need to commit to auxiliary complement advice, thereby yielding a significantly shorter witness. Moreover, by natively supporting inner-product relations, WNILA strictly generalizes IPA, enabling seamless integration with prior works. As a result, prior elegant ideas from ZGSX23 and BulletCT can be integrated within our framework. The ability to enforce linear relations further enables more sophisticated blinding techniques, which are leveraged by UniBit^{+} to reduce the signature size by 0.125 KB.

Building on WNILA, we design ZKPs that prove binary, Hamming-weight, and reciprocal constraints over committed vectors. This protocol is compressed using WNILA, in direct analogy to how the IPA compresses Bulletproof's linear-communication ZKP. The resulting construction naturally captures the K-Weight constraint of ZGSX23 and the K-Link constraint of BulletCT. Concretely, the K-Weight constraint requires proving that a committed length-N vector $\mathbf{n}$ is binary and has Hamming weight K, while the K-Link constraint reduces to proving that two committed vectors of length K are reciprocal to each other. Both types of constraints are handled natively within our protocol.

The design described so far does not, by itself, address the KKDL issue identified by Lin et al. [22] and Balla and Chaidos [5]. To this end, we incorporate the KKDL-Embedding technique of Omniring [20] to safely handle potentially adversarially generated group elements. We observe that KKDL-

Embedding is algebraically compatible with both the K-Weight technique of ZGSX23 and the K-Link technique of BulletCT. This requires only a minor modification based on a compressed Σ-protocol, analogous to the refinement introduced by Lin et al. [22] to address a flaw in the original version of Omniring.

Roadmap. In Section 2, we present the necessary preliminaries. In Section 3, we introduce WNILA as our compression pivot. In Section 4, we present the zero-knowledge proof employed by UniBit, built upon WNILA. In Section 5, we show how to integrate the KKDL-Embedding technique with our zero-knowledge proof to construct the UniBit protocol. In Section 6, we present our experimental results, and conclude our work in Section 7.

2 Preliminaries

Notation. Throughout this paper, we work over a prime-order cyclic group $\mathbb{G}$ of order $p = \Omega(2^\lambda)$, where λ denotes the security parameter, and $\mathbb{Z}_p$ denotes the corresponding scalar field. The identity element of $\mathbb{G}$ is denoted by $1_\mathbb{G}$. We use negl to denote a negligible function in λ, and poly to denote a polynomially bounded function in λ.

We always index vectors from 0. We write boldface lowercase letters (e.g., $\mathbf{a}, \mathbf{b}, \mathbf{n}$) for vectors over $\mathbb{Z}_p$, and boldface uppercase letters (e.g., $\mathbf{G}, \mathbf{A}$) for vectors of group elements. For an integer N, we use $[\mathrm{N}]$ to denote the index set $\{0, \ldots, \mathrm{N}-1\}$. The inner product of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^\mathrm{N}$ is denoted by $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=0}^{\mathrm{N}-1} a_i b_i$, and the weighted norm is defined as $\|\mathbf{n}\|_\mu^2 = \langle \mathbf{n}, \mathbf{n} \rangle_\mu = \sum_{i=0}^{\mathrm{N}-1} \mu^{i+1} n_i^2$ for a scalar $\mu \in \mathbb{Z}_p$. We also use $\mathbf{k}^n$ to denote $(k^0, \ldots, k^{n-1})$.

For a vector of group elements $\mathbf{G} = (G_0, \ldots, G_{\mathrm{N}-1})$ and a vector of scalars $\mathbf{n} = (n_0, \ldots, n_{\mathrm{N}-1})$, we use $\mathbf{G}^\mathbf{n}$ to denote the corresponding MSM $\prod_{i=0}^{\mathrm{N}-1} G_i^{n_i}$. For a scalar $e \in \mathbb{Z}_p$, the notation $\mathbf{G}^e$ denotes $(G_0^e, \ldots, G_{\mathrm{N}-1}^e)$. We use $\circ$ to denote component-wise operations. For instance, $\mathbf{G}^{\circ \mathbf{n}}$ denotes the component-wise group scalar multiplication $(G_0^{n_0}, \ldots, G_{\mathrm{N}-1}^{n_{\mathrm{N}-1}})$ and $\mathbf{G} \circ \mathbf{H} = (G_0 \cdot H_0, \ldots, G_{n-1} \cdot H_{n-1})$. The "inner product" over vectors of $\mathbb{G}$ is defined naturally: $\mathbf{G} \cdot \mathbf{H} = \prod_{i=0}^{n-1} G_i \cdot H_i$. Finally, $[\mathbf{a}]_0$ and $[\mathbf{a}]_1$ denote the subvectors consisting of the even and odd indexed entries of $\mathbf{a}$.

2.1 Computational Hardness Assumption

Our work relies on two assumptions that are equivalent to the DL assumption: the non-trivial discrete logarithm relation assumption and the interactive discrete logarithm assumption [20]. We leave the formal definitions in the full version of the paper.

2.2 Zero-Knowledge Argument of Knowledge

We assume readers are familiar with the basic notions of zero-knowledge argument of knowledge. We use *computational witness-extended emulation* (CWEE) [15,23] for knowledge soundness and *special honest-verifier zero-knowledge* (sHVZK) for zero-knowledge.

2.3 KKDL-Embedding

KKDL-Embedding [20, 22] enables Bulletproof-style proofs to safely handle adversarial group elements. To prove the constraints $\mathcal{C}(\mathbf{w}) = 1$ and $\mathbf{A}^\mathbf{w} \cdot \mathbf{B}^\mathbf{c} = 1_\mathbb{G}$ for public $\mathbf{c}$, the verifier samples fresh generators $\mathbf{G}$ and a random challenge $e \in \mathbb{Z}_p$. Defining $C = \mathbf{G}^\mathbf{w}$ and $C' = C \cdot (\mathbf{B}^\mathbf{c})^{-e}$, this yields an equivalent commitment $C' = \widehat{\mathbf{G}}^\mathbf{w}$ under $\widehat{\mathbf{G}} = \mathbf{G} \circ \mathbf{A}^e$. The parties then run any Bulletproof over $\widehat{\mathbf{G}}$. Lin et al. [22]'s "outer protocol" has a more detailed description of the technique. It is essentially a "randomized change of generator" trick that exploits the interactive DL assumption [20].

2.4 RingCT Signature

We adopt the same formulation as in BulletCT [28], where linkability is enforced via a permutation constraint. A RingCT signature scheme is modeled as a signature of knowledge for a relation $\mathbf{R}_\mathrm{RingCT}$ over an instance x and a witness w, where the instance consists of public keys, input commitments, tags, and output commitments; namely, $\mathsf{x} = (\mathbf{pk}, \mathbf{ct}, \mathbf{tg}, \mathbf{rct})$ with $\mathbf{pk} = (\mathsf{pk}_i)_{i \in [\mathrm{N}]}$, $\mathbf{ct} = (\mathsf{ct}_i)_{i \in [\mathrm{N}]}$, $\mathbf{tg} = (\mathsf{tg}_i)_{i \in [\mathrm{K}]}$, and $\mathbf{rct} = (\mathsf{rct}_i)_{i \in [\mathrm{M}]}$.

The witness encodes the real signers, the corresponding secret keys, commitment openings, tag secrets, and receiver openings: $\mathsf{w} = (\phi, \mathbf{sk}_\phi, \mathbf{a}_\phi, \mathbf{r}_\phi, \mathbf{s}, \mathbf{a}', \mathbf{r}')$. The signature of knowledge establishes that, for a secret injection $\phi : [\mathrm{K}] \hookrightarrow [\mathrm{N}]$,

- **K-out-of-N.** The prover knows $\mathbf{sk}_\phi, \mathbf{a}_\phi, \mathbf{r}_\phi \in \mathbb{Z}_p^\mathrm{K}$ such that for all $i \in [\mathrm{K}]$, $\mathsf{pk}_{\phi(i)} = G^{\mathsf{sk}_{\phi(i)}}$ and $\mathsf{ct}_{\phi(i)} = W^{a_{\phi(i)}} E^{r_{\phi(i)}}$.
- **Linkability.** The prover knows $\mathbf{s} \in \mathbb{Z}_p^\mathrm{K}$ such that for all $i \in [\mathrm{K}]$, $\mathsf{tg}_i = Q^{s_i}$ and $\mathbf{s}$ is a permutation of $\mathbf{sk}_\phi$.
- **Range.** The prover knows $\mathbf{a}', \mathbf{r}' \in \mathbb{Z}_p^\mathrm{M}$ such that for all $i \in [\mathrm{M}]$, $\mathsf{rct}_i = W^{a'_i} E^{r'_i}$ with $a'_i \in [0, 2^\beta - 1]$.
- **Balance.** The input and output amounts satisfy $\sum_{i \in [\mathrm{K}]} a_{\phi(i)} = \sum_{i \in [\mathrm{M}]} a'_i$.

3 Weighted Norm Inner Linear Argument

We now formalize the relation captured by the weighted norm inner linear argument. A WNILA is an argument of knowledge for the following relation:

$$\mathbf{R}_\mathrm{WNILA} = \left\{ \begin{array}{l} U, \mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H}, C, \\ \mathbf{c}, \psi, \mu = \psi^2; \\ \mathbf{n}, \mathbf{a}, \mathbf{b}, \mathbf{l}, v \end{array} \middle| \begin{array}{l} C = U^v \mathbf{G}^\mathbf{n} \mathbf{A}^\mathbf{a} \mathbf{B}^\mathbf{b} \mathbf{H}^\mathbf{l}, \\ v = \|\mathbf{n}\|_\mu^2 + \langle \mathbf{a} \| \mathbf{c}, \mathbf{b} \| \mathbf{l} \rangle \end{array} \right\}$$

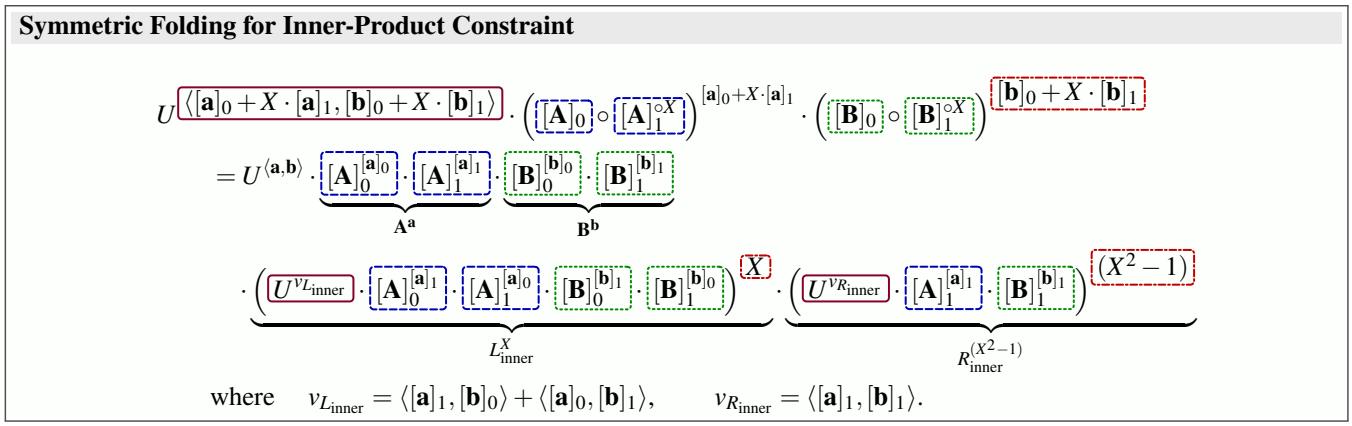

Figure 3: The algebraic structure underlying the symmetric folding of the inner-product component of WNILA. The weighted-norm and linear components follow analogous relations. In particular, the aggregate coefficient of the X^2 basis is separated from the (X^2-1) and 1 bases, where the 1 basis compensates for the target commitment C.

where $\mathbf{n} \in \mathbb{Z}_p^{\mathrm{N}}$, $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^{\mathrm{K}}$, $\mathbf{c}, \mathbf{l} \in \mathbb{Z}_p^{\ell}$, $\mathbf{G} \in \mathbb{G}^{\mathrm{N}}$, $\mathbf{A}, \mathbf{B} \in \mathbb{G}^{\mathrm{K}}$ and $\mathbf{H} \in \mathbb{G}^{\ell}$. We assume that computing a non-trivial discrete-logarithm relation among $(U, \mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H})$ is computationally infeasible. Our construction is inspired by the Weighted-Norm Linear Argument (WNLA) in Bulletproof^{++} [12], but additionally supports inner-product constraints.

Below we make the folding structure explicit. Let $\mathbb{G}[X]$ denote formal polynomials with coefficients in $\mathbb{G}$. We encode $A_0 + A_1 X + \cdots + A_d X^d$ as the multiplicative expression $A_0 \cdot A_1^X \cdots A_d^{X^d}$. Under this notation, each WNILA round corresponds to a polynomial identity check in $\mathbb{G}[X]$ that reduces the instance size by a factor of two upon evaluation at a random challenge. Our WNILA construction employs the symmetric folding technique. Concretely, we consider the following *folding polynomials*:

$$\mathbf{G}'(X) = [\mathbf{G}]_0^{\circ \psi} \circ [\mathbf{G}]_1^{\circ X}, \quad \mathbf{A}'(X) = [\mathbf{A}]_0 \circ [\mathbf{A}]_1^{\circ X}, \tag{1}$$

$$\mathbf{B}'(X) = [\mathbf{B}]_0 \circ [\mathbf{B}]_1^{\circ X}, \quad \mathbf{H}'(X) = [\mathbf{H}]_0 \circ [\mathbf{H}]_1^{\circ X}, \tag{2}$$

$$\mathbf{c}'(X) = [\mathbf{c}]_0 + X \cdot [\mathbf{c}]_1, \quad \mu' = \mu^2, \tag{3}$$

$$\mathbf{n}'(X) = \psi^{-1} [\mathbf{n}]_0 + X \cdot [\mathbf{n}]_1, \tag{4}$$

$$\mathbf{a}'(X) = [\mathbf{a}]_0 + X [\mathbf{a}]_1, \quad \mathbf{b}'(X) = [\mathbf{b}]_0 + X [\mathbf{b}]_1, \tag{5}$$

$$\mathbf{l}'(X) = [\mathbf{l}]_0 + X [\mathbf{l}]_1 \tag{6}$$

$$\mathbf{v}'(X) = \|\mathbf{n}'(X)\|_{\mu'}^2 + \langle \mathbf{a}'(X), \mathbf{b}'(X) \rangle + \langle \mathbf{c}'(X), \mathbf{l}'(X) \rangle \tag{7}$$

Applying the WNILA constraint directly to the folding polynomials, we can verify that

$$U^{v'(X)} \cdot \mathbf{G}'(X)^{\mathbf{n}'(X)} \cdot \mathbf{A}'(X)^{\mathbf{a}'(X)} \cdot \mathbf{B}'(X)^{\mathbf{b}'(X)} \cdot \mathbf{H}'(X)^{\mathbf{l}'(X)} = C \cdot L^X \cdot R^{(X^2-1)} \tag{8}$$

where $L, R \in \mathbb{G}$ are the group elements that do not depend on X. Concretely, they are computed as follows: $S_1 = \{(\mathbf{a}, \mathbf{A}), (\mathbf{b}, \mathbf{B}), (\mathbf{l}, \mathbf{H})\}$, $S_2 = \{(\mathbf{a}, \mathbf{b}), (\mathbf{c}, \mathbf{l})\}$.

$$L = U^{v_L} \cdot [\mathbf{G}]_1^{\psi^{-1} \cdot [\mathbf{n}]_0} \cdot [\mathbf{G}]_0^{\psi \cdot [\mathbf{n}]_1} \cdot \prod_{(\mathbf{t}, \mathbf{T}) \in S_1} [\mathbf{T}]_0^{[\mathbf{t}]_1} [\mathbf{T}]_1^{[\mathbf{t}]_0}, \tag{9}$$

$$v_L = 2\psi^{-1} \langle [\mathbf{n}]_0, [\mathbf{n}]_1 \rangle_{\mu^2} + \sum_{(\mathbf{x}, \mathbf{y}) \in S_2} \left(\langle [\mathbf{x}]_0, [\mathbf{y}]_1 \rangle + \langle [\mathbf{x}]_1, [\mathbf{y}]_0 \rangle \right), \tag{10}$$

$$R = U^{v_R} \cdot [\mathbf{G}]_1^{[\mathbf{n}]_1} \cdot \prod_{(\mathbf{t}, \mathbf{T}) \in S_1} [\mathbf{T}]_1^{[\mathbf{t}]_1}, \tag{11}$$

$$v_R = \|[\mathbf{n}]_1\|_{\mu^2}^2 + \sum_{(\mathbf{x}, \mathbf{y}) \in S_2} \langle [\mathbf{x}]_1, [\mathbf{y}]_1 \rangle. \tag{12}$$

Equation (8) can be verified by evaluating it at a random $\zeta \in \mathbb{Z}_p$, yielding a WNILA instance with parameters $(\mathrm{N}/2, \mathrm{K}/2, \ell/2)$. Figure 3 illustrates this symmetric folding for the inner-product component; the weighted-norm and linear components follow the same algebraic pattern. Based on this observation, we obtain a recursive self-reduction protocol. The prover sends commitments L and R, and the verifier samples $\zeta \leftarrow_{\$} \mathbb{Z}_p$. Both parties then update the commitment as $C' = C \cdot L^{\zeta} \cdot R^{\zeta^2 - 1}$, with generators folded according to (1)–(7), while the prover privately updates the witness.

Theorem 1. *WNILA achieves perfect completeness. Assuming finding a non-trivial discrete logarithm relation among $(\mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H}, U)$ is computationally infeasible, WNILA satisfies CWEE.*

A formal proof is deferred to the full version of the paper.

Efficiency. The proof size consists of $2\log(\max(\mathrm{N}, \mathrm{K}, \ell))$ group elements and 4 field elements. Verification is dominated by a single MSM of size $\mathrm{N} + 2\mathrm{K} + \ell + 1$. At the i-th folding round, the prover performs two MSMs of sizes $\mathrm{N}/2^i + \frac{3}{2}\mathrm{K}/2^i$ and $\frac{1}{2}(\mathrm{N}/2^i + \mathrm{K}/2^i)$, respectively, together

Public input: $C \in \mathbb{G}$, $(\mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H}) \in \mathbb{G}^{N+2K+\ell}$, $\mathbf{c} \in \mathbb{Z}_p^\ell$, $\mu, \psi \in \mathbb{Z}_p$ and $\mu = \psi^2$.
Prover input: $(\mathbf{n}, \mathbf{a}, \mathbf{b}, \mathbf{l}) \in \mathbb{Z}_p^{N+2K+\ell}$.

$\mathcal{P} \Rightarrow \mathcal{V}$: Send $L, R \in \mathbb{G}$ according to (9) and (11)

$\mathcal{V} \Rightarrow \mathcal{P}$: Send $\zeta \leftarrow_\$ \mathbb{Z}_p$. This challenge is the "evaluation challenge" for the polynomial identity check over $\mathbb{G}[X]$ defined in equation (8).

$\mathcal{P}$ and $\mathcal{V}$: Compute the folded generators and the public parameters $\mathbf{G}' = \mathbf{G}(\zeta)$, $\mathbf{A}' = \mathbf{A}(\zeta)$, $\mathbf{B}' = \mathbf{B}(\zeta)$, $\mathbf{H}' = \mathbf{H}(\zeta)$, $\mathbf{c}' = \mathbf{c}(\zeta)$, $\mu' = \mu^2$ according to the equations (1) $\sim$ (3), and compute the reduced target commitment $C' = C \cdot L^{\zeta} \cdot R^{\zeta^2 - 1}$.

$\mathcal{P}$: Compute the folded witness $\mathbf{n}' = \mathbf{n}(\zeta)$, $\mathbf{a}' = \mathbf{a}(\zeta)$, $\mathbf{b}' = \mathbf{b}(\zeta)$, $\mathbf{l}' = \mathbf{l}(\zeta)$ according to the equations (4) $\sim$ (6).

$\mathcal{P} \Leftrightarrow \mathcal{V}$: Repeatedly execute the same procedure on the reduced instance $(U, \mathbf{G}', \mathbf{A}', \mathbf{B}', \mathbf{H}', \mathbf{c}', \mu', C')$ and the reduced witness $(\mathbf{n}', \mathbf{a}', \mathbf{b}', \mathbf{l}')$. This process continues until the instance size becomes small enough, at which point the prover reveals the remaining witness directly.

Figure 4: Self-folding procedure of WNILA.

with $N/2^{i+1} + K/2^i$ additional group scalar operations arising from generator folding. In practice, these group scalar operations dominate the prover cost.

Treating v as a public value. In our application, v must be treated as a public value:

$$\mathbf{R}_{\text{WNILA}'} = \left\{ \begin{array}{l} U, \mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H}, C, \\ v, \mathbf{c}, \psi, \mu = \psi^2; \\ \mathbf{n}, \mathbf{a}, \mathbf{b}, \mathbf{l} \end{array} \middle| \begin{array}{l} C = \mathbf{G}^{\mathbf{n}} \mathbf{A}^{\mathbf{a}} \mathbf{B}^{\mathbf{b}} \mathbf{H}^{\mathbf{l}}, \\ v = \|\mathbf{n}\|_\mu^2 + \langle \mathbf{a} \,\|\, \mathbf{c}, \mathbf{b} \,\|\, \mathbf{l} \rangle \end{array} \right\}$$

in which v is not committed in C. This variant can be realized via a simple wrapper, following the same approach as Protocol 1 of Bulletproof [8]: The verifier samples a random challenge x, and both parties define the modified commitment $\widehat{C} = C \cdot U^{xv}$. The instance $(\widehat{C}, \mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H}, U^x)$ is then handled as a standard WNILA instance for the extended witness $(v, \mathbf{n}, \mathbf{a}, \mathbf{b}, \mathbf{l})$.

Handling non-uniform dimensions. In typical RingCT settings, one may have $N > K > \ell$ or $N > \ell > K$. Consider the former case. During the recursive reduction, the protocol eventually reaches $\ell = 1$, at which point the instance contains a single group element H', a single scalar c', and a single witness scalar l'. The prover reveals l', and both parties update the commitment according to $C' = C \cdot H'^{-l'} \cdot U^{-c'l'}$. This eliminates the linear component from all subsequent steps.

Generalized to δ-fold-WNILA. The protocol extends naturally to a generalized relation, denoted $\mathbf{R}_{\delta\text{-fold-WNILA}}$, over public inputs $U, (\mathbf{G}_i)_{i \in [\delta]}, \mathbf{A}, \mathbf{B}, \mathbf{H}, C, \mathbf{c}, \boldsymbol{\psi}, \boldsymbol{\mu}$ and witnesses $(\mathbf{n}_i)_{i \in [\delta]}, \mathbf{a}, \mathbf{b}, \mathbf{l}$, where $\mu_i = \psi_i^2$, and the relation enforces that $C = U^v \prod_{i \in [\delta]} \mathbf{G}_i^{\mathbf{n}_i} \mathbf{A}^{\mathbf{a}} \mathbf{B}^{\mathbf{b}} \mathbf{H}^{\mathbf{l}}$ with $v = \sum_{i \in [\delta]} \|\mathbf{n}_i\|_{\mu_i}^2 + \langle \mathbf{a} \,\|\, \mathbf{c}, \mathbf{b} \,\|\, \mathbf{l} \rangle$.

The standard WNILA corresponds to the special case $\delta = 1$. The protocol is analogous to WNILA, as the same folding structure is applied independently to each $(\mathbf{G}_i, \mathbf{n}_i, \mu_i)$. The number of folding rounds is $\log(\max(|\mathbf{n}_i|, K, \ell))$, and the proof consists of $2\log(\max(|\mathbf{n}_i|, K, \ell))$ group elements and $\delta + 3$ field elements. The $\mathbf{R}_{\text{2-fold-WNILA}}$ instance is used in UniBitRange, where the two weighted norms correspond to the N binary constraints from the K-Weight constraint and $\beta \cdot M$ binary constraints from M range-proofs.

4 Zero-Knowledge Protocol

In this section, we present a zero-knowledge protocol for enforcing binary, Hamming-weight, and reciprocal constraints.

$$\mathbf{R}_{\text{inner}} = \left\{ \begin{array}{l} C', \widehat{\mathbf{G}}, \widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{H}}, \gamma; \\ \mathbf{n}, \boldsymbol{\alpha}, \boldsymbol{\rho}, \mathbf{l} \end{array} \middle| \begin{array}{l} C' = \widehat{\mathbf{G}}^{\mathbf{n}} \widehat{\mathbf{A}}^{\boldsymbol{\alpha}} \widehat{\mathbf{B}}^{\boldsymbol{\rho}} \widehat{\mathbf{H}}^{\mathbf{l}}, \\ \mathbf{n} \in \{0,1\}^N, \\ \sum_i n_i = K, \\ (\boldsymbol{\alpha} + \gamma \cdot \mathbf{1}^K) \circ \boldsymbol{\rho} = \mathbf{1}^K \end{array} \right\}.$$

Let $\mathbf{a} = \boldsymbol{\alpha} + \gamma \cdot \mathbf{1}^K$ and $\mathbf{b} = \boldsymbol{\rho}$. The protocol needs to prove the following constraints: $\mathbf{a} \circ \mathbf{b} = \mathbf{1}^K$, $(\mathbf{n} - \frac{1}{2} \cdot \mathbf{1}^N)^2 = \frac{1}{4} \cdot \mathbf{1}^N$, and $\sum_{i=0}^{N-1} n_i = K$. These constraints can be compressed into a single constraint using challenges μ, κ, and η.

$$\begin{aligned} &\|\mathbf{n} - \tfrac{1}{2}\mathbf{1}^N\|_\mu^2 - 2\kappa \sum_{i=0}^{N-1} n_i + \kappa^2 \langle \boldsymbol{\eta}^K \circ \mathbf{b}, \mathbf{a} \rangle \\ &= \tfrac{1}{4} \sum_{i=0}^{N-1} \mu^{i+1} - 2\kappa K + \kappa^2 \sum_{i=0}^{K-1} \eta^i. \end{aligned} \tag{16}$$

To achieve zero-knowledge, the prover masks the witness vectors using random blinding values $\mathbf{s}_G, \mathbf{s}_A, \mathbf{s}_B$, and $\mathbf{s}_H$, and encodes the masked witnesses as low-degree polynomials:

$$\mathbf{n}(X) = \mathbf{n} - \tfrac{1}{2}\mathbf{1}^N - \kappa \mu^{-1} \boldsymbol{\mu}^{-N} + \mathbf{s}_G \cdot X, \tag{17}$$

$$\mathbf{a}(X) = \mathbf{a} + \mathbf{s}_A \cdot X, \tag{18}$$

$$\mathbf{b}(X) = \kappa^2 \big(\boldsymbol{\eta}^K \circ (\mathbf{b} + \mathbf{s}_B \cdot X) \big), \tag{19}$$

$$\mathbf{l}(X) = \mathbf{l} + \mathbf{s}_H \cdot X. \tag{20}$$

With these encodings, the combined constraint admits a quadratic polynomial representation of the form

$$\|\mathbf{n}(X)\|_\mu^2 + \langle \mathbf{a}(X), \mathbf{b}(X) \rangle = t_0 + t_1 X + t_2 X^2, \tag{21}$$

where

$$t_0 = \frac{1}{4} \sum_{i=0}^{N-1} \mu^{i+1} - 2 \cdot \kappa \cdot K + \kappa^2 \cdot \sum_{i=0}^{K-1} \eta^i + \delta(\kappa, \mu) \tag{22}$$

for $\delta(\kappa, \mu) = \kappa \cdot N + \kappa^2 \cdot \sum_{i=0}^{N-1} \mu^{-i-1}$. The protocol is analogous to Bulletproof's linear ZK protocol, which essentially tests the polynomial identity equation (21), in which the verifier can symbolically compute the term t_0 on its own. A description of this ZK protocol is provided in Figure 5.

Public input: $C \in \mathbb{G}$, $(\widehat{\mathbf{G}}, \widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{H}}) \in \mathbb{G}^{\mathrm{N}+2\mathrm{K}+\ell}$, $\gamma \in \mathbb{Z}_p$.
Prover input: $(\mathbf{n}, \boldsymbol{\alpha}, \boldsymbol{\rho}, \mathbf{l}) \in \mathbb{Z}_p^{\mathrm{N}+2\mathrm{K}+\ell}$.

$\mathcal{P} \Rightarrow \mathcal{V}$: Commit to the blinding $S = \widehat{\mathbf{G}}^{\mathbf{s}_G} \cdot \widehat{\mathbf{A}}^{\mathbf{s}_A} \cdot \widehat{\mathbf{B}}^{\mathbf{s}_B} \cdot \widehat{\mathbf{H}}^{\mathbf{s}_H}$ where $(\mathbf{s}_G, \mathbf{s}_A, \mathbf{s}_B, \mathbf{s}_H) \leftarrow_\$ \mathbb{Z}_p^{\mathrm{N}+2\mathrm{K}+\ell}$.

$\mathcal{V} \Rightarrow \mathcal{P}$: Send the random challenges $\psi, \kappa, \eta \leftarrow_\$ \mathbb{Z}_p$ and define $\mu = \psi^2$.

$\mathcal{P} \Rightarrow \mathcal{V}$: Commit to the degree-1 and degree-2 coefficients of the polynomial defined by (21) as $T_i = W^{t_i} \cdot E^{\tau_i}, i \in \{1,2\}$, for which $(\tau_1, \tau_2) \leftarrow_\$ \mathbb{Z}_p^2$.

$\mathcal{V} \Rightarrow \mathcal{P}$: Send a random challenge $x \leftarrow_\$ \mathbb{Z}_p$. This is an evaluation challenge for the polynomial identity equation (21).

$\mathcal{P} \Rightarrow \mathcal{V}$: Send $\mathbf{n}' = \mathbf{n}(x)$, $\mathbf{a}' = \mathbf{a}(x)$, $\mathbf{b}' = \mathbf{b}(x)$, $\mathbf{l}' = \mathbf{l}(x)$, defined according to (17) $\sim$ (20), and also send the blinding offset $\tau_x = \tau_1 x + \tau_2 x^2$, along with the correct polynomial evaluation value $\hat{t} = \|\mathbf{n}'\|_\mu^2 + \langle \mathbf{a}', \mathbf{b}' \rangle$.

$\mathcal{V}$: Define

$$Y = C' \cdot \widehat{\mathbf{G}}^{-\frac{1}{2}\mathbf{1}^{\mathrm{N}} - \kappa\mu^{-1}\boldsymbol{\mu}^{-\mathrm{N}}} \cdot \widehat{\mathbf{A}}^{\gamma \cdot \mathbf{1}^{\mathrm{K}}} \cdot S^x,$$
$$\widehat{\mathbf{B}}' = \widehat{\mathbf{B}}^{\circ \kappa^{-2} \cdot \boldsymbol{\eta}^{-\mathrm{K}}},$$
$$c = \delta(\kappa, \mu) + \tfrac{1}{4} \cdot \sum_{i=0}^{\mathrm{N}-1} \mu^{i+1} - 2 \cdot \kappa \cdot \mathrm{K} + \kappa^2 \cdot \sum_{i=0}^{\mathrm{K}-1} \eta^i.$$

Y is the commitment to the $\mathbf{n}', \mathbf{a}', \mathbf{b}', \mathbf{l}'$ computed through homomorphic evaluation of the equations (17) $\sim$ (20) in the commitment space $\mathbb{G}$.
Accept if and only if all of the following equations hold:

$$Y \stackrel{?}{=} \widehat{\mathbf{G}}^{\mathbf{n}'} \widehat{\mathbf{A}}^{\mathbf{a}'} \widehat{\mathbf{B}}'^{\mathbf{b}'} \widehat{\mathbf{H}}^{\mathbf{l}'}, \tag{13}$$
$$W^{\hat{t}} E^{\tau_x} \stackrel{?}{=} W^c \cdot T_1^x \cdot T_2^{x^2}, \tag{14}$$
$$\hat{t} \stackrel{?}{=} \|\mathbf{n}'\|_\mu^2 + \langle \mathbf{a}', \mathbf{b}' \rangle. \tag{15}$$

(13) checks if $\mathbf{n}'$, $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{l}'$ are computed correctly through the polynomial encoding equations (17) $\sim$ (20). (14) checks if the commitment to the degree-1 and degree-2 coefficients of the polynomial is computed correctly. (15) checks if the polynomial evaluation value $\hat{t}$ is computed correctly.

Figure 5: Bulletproof-style ZK protocol for $\mathbf{R}_{\text{inner}}$.

Theorem 2. *The protocol is an argument of knowledge for* $\mathbf{R}_{\text{inner}}$ *with perfect completeness, CWEE, and sHVZK.*

A formal proof is deferred to the full version of the paper.

4.1 Compression via WNILA

The "linear communication portion" of this ZKP can be compressed to logarithmic communication using WNILA. Instead of sending the blinded witnesses $\mathbf{n}', \mathbf{a}', \mathbf{b}', \mathbf{l}'$ explicitly, the prover and verifier invoke the wrapped WNILA protocol for $\mathbf{R}_{\text{WNILA}'}$. Concretely, this protocol is instantiated with $v = \hat{t}$, generators $\widehat{\mathbf{G}}, \widehat{\mathbf{A}}, \widehat{\mathbf{B}}', \widehat{\mathbf{H}}$, commitment Y, witness $(\mathbf{n}', \mathbf{a}', \mathbf{b}', \mathbf{l}')$, and a trivial linear vector $\mathbf{c} = \mathbf{0}^8$. This reduces the communication complexity from $O(\mathrm{N})$ to $O(\log \mathrm{N})$.

5 The UniBit Protocol

In this section, we combine our tools to construct UniBit. We present the protocol in Figure 6 and explain its construction in the following three subsections.

5.1 Establishing a Secure Commitment

The goal of the Σ-protocol is to provide the verifier with a Pedersen vector commitment to a collection of witness values used in the remainder of the protocol. Concretely, at the end of KKDL-Embedding, both parties establish a secure commitment to the following witness vector:

$$\left(\mathbf{n},\ \boldsymbol{\alpha},\ \boldsymbol{\rho},\ w_{\mathsf{sk}},\ w_a,\ w_r,\ w_\Delta\right) \in \mathbb{Z}_p^{\mathrm{N}+2\mathrm{K}+4}. \tag{23}$$

The components of this vector play different roles in the protocol. In particular, $\mathbf{n}$ is required to remain fixed across all verifier challenges, whereas $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ are derived from the verifier challenges v and (v, γ), respectively. Concretely, $\mathbf{n}$ encodes the secret injection ϕ in binary form:

$$n_i = 1 \iff \exists\, j \in [\mathrm{K}] : \phi(j) = i, \quad n_i = 0 \text{ otherwise.} \tag{24}$$

$\boldsymbol{\alpha}$ is determined by a random verifier challenge v as

$$\boldsymbol{\alpha} = (v^{\phi(0)}, \ldots, v^{\phi(\mathrm{K}-1)}), \tag{25}$$

and $\boldsymbol{\rho}$ is derived from an additional challenge γ:

$$\boldsymbol{\rho} = \left(1/(v^{\phi(0)} + \gamma), \ldots, 1/(v^{\phi(\mathrm{K}-1)} + \gamma)\right) \tag{26}$$

and $w_{\mathsf{sk}} = \sum_{i=0}^{\mathrm{K}-1} v^{\phi(i)} \cdot \mathsf{sk}_{\phi(i)}$, $w_a = \sum_{i=0}^{\mathrm{K}-1} v^{\phi(i)} \cdot a_{\phi(i)}$, $w_r = \sum_{i=0}^{\mathrm{K}-1} v^{\phi(i)} \cdot r_{\phi(i)}$, $w_\Delta = \sum_{i=0}^{\mathrm{K}-1} r_{\phi(i)} - \sum_{i=0}^{\mathrm{M}-1} r'_i$.

We use an amortized compressed Σ-protocol [4] (Appendix E, Protocol 9) to ensure that the commitments V and P are correctly formed as

$$V = \mathbf{A}^{\boldsymbol{\alpha}} H_0^{w_{\mathsf{sk}}} H_1^{w_a} H_2^{w_r} H_3^{w_\Delta} \mathbf{H}_{4:7}^{\mathbf{r}_V}, \qquad P = \mathbf{B}^{\boldsymbol{\rho}} \mathbf{H}_{4:}^{\mathbf{r}_P},$$

where $\mathbf{r}_V \leftarrow_\$ \mathbb{Z}_p^4$. For this reason, we set $r_{V,2} = 0$ and $r_{P,1} = 0$, which aligns with the interface of the protocol provided by [4]. This step ensures that the prover cannot cheat by adversarially altering the committed bit-vector $\mathbf{n}$ by sending malformed commitments V or P. From the verifier's perspective, after the Σ-protocol, the relevant witness components are bound with the correct dependency structure. Note that when K is small, one can instead use a classic 3-move Σ-protocol to get better communication efficiency and a slightly faster signing and verification.

Public input: public keys $\mathbf{pk} \in \mathbb{G}^{\mathrm{N}}$, input coin commitments $\mathbf{ct} \in \mathbb{G}^{\mathrm{N}}$, tags $\mathbf{tg} \in \mathbb{G}^{\mathrm{K}}$, output coin commitments $\mathbf{rct} \in \mathbb{G}^{\mathrm{M}}$, and generators $W, E, F, Q \in \mathbb{G}^4$. Both parties also agree on a separate DL-based range-proof scheme (e.g., Bulletproof^{++} [12]), where ε is the number of generators required by this range-proof scheme.
Prover input: secret keys $\mathbf{sk}_\phi \in \mathbb{Z}_p^{\mathrm{K}}$, openings of the input coin commitments $\mathbf{a}_\phi, \mathbf{r}_\phi \in \mathbb{Z}_p^{\mathrm{K}}$, openings of the output coin commitments $\mathbf{a}', \mathbf{r}' \in \mathbb{Z}_p^{\mathrm{M}}$, the secret injection $\phi : [\mathrm{K}] \hookrightarrow [\mathrm{N}]$.

$\mathcal{V} \Rightarrow \mathcal{P}$: Sample $(\mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{H}, U) \leftarrow_\$ \mathbb{G}^{\mathrm{N}+2\mathrm{K}+8+1}$ and if $\varepsilon > (\mathrm{N}+2\mathrm{K}+8+1)$ additionally sample $\mathbf{G}' \leftarrow_\$ \mathbb{G}^{\varepsilon-(\mathrm{N}+2\mathrm{K}+8+1)}$ and send them to the prover. These additional generators $\mathbf{G}'$ ensure that *the range-proof scheme* has enough generators to work.

$\mathcal{P} \Rightarrow \mathcal{V}$: Commit to $\mathbf{n}$ as $B = \mathbf{G}^{\mathbf{n}}\mathbf{H}_{4:}^{\mathbf{r}_B}, \mathbf{r}_B \leftarrow_\$ \mathbb{Z}_p^4$ where $\mathbf{n}$ is defined in (24). This step binds $\mathbf{n}$ at the very beginning.

$\mathcal{V} \Rightarrow \mathcal{P}$: Send random challenge $v \leftarrow_\$ \mathbb{Z}_p$.

$\mathcal{P} \Rightarrow \mathcal{V}$: Commit to the witness vector that depends on v as $V = \mathbf{A}^{\boldsymbol{\alpha}} H_0^{w_{\mathsf{sk}}} H_1^{w_a} H_2^{w_r} H_3^{w_\Delta} \mathbf{H}_{4:7}^{\mathbf{r}_V}, \mathbf{r}_V \leftarrow_\$ \mathbb{Z}_p^4$, with $r_{V,2} = 0$. These witness values are defined in (23).

$\mathcal{V} \Rightarrow \mathcal{P}$: Send the random challenge $\gamma \leftarrow_\$ \mathbb{Z}_p$.

$\mathcal{P} \Rightarrow \mathcal{V}$: Commit to the witness vector that depends on γ as $P = \mathbf{B}^{\boldsymbol{\rho}} \mathbf{H}_{4:}^{\mathbf{r}_P}, \mathbf{r}_P \leftarrow_\$ \mathbb{Z}_p^4$, with $r_{P,1} = 0$.

$\mathcal{P} \Leftrightarrow \mathcal{V}$: Execute an amortized compressed Σ-protocol [4] to prove V and P are formed as $\mathbf{A}^{\boldsymbol{\alpha}} \cdot H_0^{w_{\mathsf{sk}}} \cdot H_1^{w_a} \cdot H_2^{w_r} \cdot H_3^{w_\Delta}$ and $\mathbf{B}^{\boldsymbol{\rho}} \cdot \mathbf{H}_{4:}^{\mathbf{r}_P}$. If K is small, simply execute a classic three-move Σ-protocol instead. If the verifier accept this protocol, then both parties define the commitment $C = B \cdot V \cdot P$.

$\mathcal{V} \Rightarrow \mathcal{P}$: Send the random challenges $e, u \leftarrow_\$ \mathbb{Z}_p$.

$\mathcal{P}$ and $\mathcal{V}$: Define $\widehat{\mathbf{G}}$, $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$, $\widehat{H}_0$, $\widehat{H}_1$, $\widehat{H}_2$, $\widehat{H}_3$, $\widehat{\mathbf{H}}_{4:7}$ and C' according to (33)–(37). This step embeds the equations (27) $\sim$ (29) into a secure commitment C'. If the prover is honest, we have $C' = \widehat{\mathbf{G}}^{\mathbf{n}} \widehat{\mathbf{A}}^{\boldsymbol{\alpha}} \widehat{\mathbf{B}}^{\boldsymbol{\rho}} \widehat{\mathbf{H}}^{\mathbf{l}}$.

$\mathcal{P} \Leftrightarrow \mathcal{V}$: Execute our ZK protocol for $\mathbf{R}_{\text{inner}}$ to prove that $\mathbf{n}$ is a binary vector with Hamming weight K and $(\boldsymbol{\alpha} + \gamma \cdot \mathbf{1}^{\mathrm{K}}) \circ \boldsymbol{\rho} = \mathbf{1}^{\mathrm{K}}$, with respect to the commitment C' under the generators $\widehat{\mathbf{G}}, \widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{H}}$ and the scalar γ.

$\mathcal{P} \Leftrightarrow \mathcal{V}$: Use the ε generators at hand to execute an aggregated range-proof scheme to prove that the opening of $\mathbf{rct}$ is within a range of $[0, 2^\beta - 1]$.

Figure 6: UniBit Protocol

5.2 Applying KKDL-Embedding Technique

The goal of KKDL-Embedding is to incorporate all group–scalar constraints of the form $\mathbf{A}^{\mathbf{w}} \cdot \mathbf{B}^{\mathbf{c}} \stackrel{?}{=} 1_{\mathbb{G}}$ into the established commitment C.

Embedding K-Weight [31] constraint. To capture the K-out-of-N constraint, we apply KKDL-Embedding to the following two group-scalar equalities induced by the K-Weight technique of ZGSX23 [31]:

$$\mathbf{pk}^{\mathbf{v}^{\mathrm{N}} \circ \mathbf{n}} \cdot G^{-w_{\mathsf{sk}}} \stackrel{?}{=} 1_{\mathbb{G}}, \tag{27}$$

$$\mathbf{ct}^{\mathbf{v}^{\mathrm{N}} \circ \mathbf{n}} \cdot W^{-w_a} \cdot E^{-w_r} \stackrel{?}{=} 1_{\mathbb{G}}. \tag{28}$$

Under the assumption that $\mathbf{n}$ is a binary vector of Hamming weight K, which is proved in our ZK protocol, these equalities guarantee the existence of a consistent random linear combination of the secret keys and the openings of the corresponding coin commitments.

Embedding balance constraint. The balance constraint is enforced by the following equation:

$$\mathbf{ct}^{\mathbf{n}} \cdot \mathbf{rct}^{-\mathbf{1}^{\mathrm{M}}} \cdot E^{-w_\Delta} \stackrel{?}{=} 1_{\mathbb{G}}. \tag{29}$$

Equality (29) ensures that the total exponent of W contributed by the selected input commitments is equal to that of the output commitments.

Embedding K-Link [28] constraint. To enforce linkability, we rely on the K-Link technique of BulletCT [28], which is formalized as Lemma 1 in that work.

Lemma 1 (Lemma 1 of BulletCT [28])**.** *Given two sequences of secret keys $(sk_{\phi(i)})_{i=0}^{\mathrm{K}-1}$ and $(s'_i)_{i=0}^{\mathrm{K}-1}$, hidden in the source public keys and tags, for an arbitrary value $v \leftarrow_\$ \mathbb{Z}_p$, if the tag exponent vector $(\alpha_i)_{i=0}^{\mathrm{K}-1}$ satisfies a "weighted sum" constraint $\sum_{i=0}^{K-1} v^{\phi(i)} sk_{\phi(i)} = \sum_{i=0}^{\mathrm{K}-1} \alpha_i s'_i$, and a permutation constraint $(\alpha_i)_{i=0}^{\mathrm{K}-1} \sim (v^{\phi(i)})_{i=0}^{\mathrm{K}-1}$, the probability that the two secret key sequences fail to satisfy a permutation relation is negligible for all PPT adversaries $\mathcal{A}$:*

$$\Pr\left[\begin{array}{l|l} \sum_{i=0}^{\mathrm{K}-1} v^{\phi(i)} sk_{\phi(i)} = \sum_{i=0}^{\mathrm{K}-1} \alpha_i s'_i, & \mathsf{crs} \leftarrow \mathcal{G}(1^\lambda), \\ (\alpha_i)_{i=0}^{\mathrm{K}-1} \sim (v^{\phi(i)})_{i=0}^{\mathrm{K}-1}, & \mathcal{A}(\mathsf{crs}) \rightarrow \\ (s_{\phi(i)})_{i=0}^{\mathrm{K}-1} \nsim (s'_i)_{i=0}^{\mathrm{K}-1} & \left(\begin{array}{c} s_{\phi(i)}, \\ s'_i \end{array}\right)_{i=0}^{\mathrm{K}-1} \end{array}\right] \leq \text{negl}.$$

Here, $\sim$ and $\nsim$ denote the permutation and non-permutation relations, respectively.

Lemma 1 implies that, in addition to enforcing the K-Weight constraint, it suffices to prove the following group-scalar equality:

$$\mathbf{tg}^{\boldsymbol{\alpha}} \cdot Q^{-w_{\mathsf{sk}}} \stackrel{?}{=} 1_{\mathbb{G}} \tag{30}$$

and additionally prove that $\boldsymbol{\alpha}$ is a permutation of $\mathbf{v}^{\phi(i)}$. We follow the approach of BulletCT [28], which applies the log-derivative technique [16] to prove that

$$\sum_{j=0}^{\mathrm{N}-1} \frac{n_j}{v^j + \gamma} = \sum_{i=0}^{\mathrm{K}-1} \frac{1}{\alpha_i + \gamma},$$

where γ is a random challenge sampled after $\boldsymbol{\alpha}$ and $\mathbf{n}$ are bound to the verifier. This is done by introducing an additional

generator $F \in \mathbb{G}$ and a vector $\boldsymbol{\rho} = \{(\alpha_i+\gamma)^{-1}\}_{i\in[\mathrm{K}]} \in \mathbb{Z}_p^{\mathrm{K}}$, and proving that

$$\left(F^{\circ(\mathbf{v}^{\mathrm{N}}+\gamma\cdot\mathbf{1}^{\mathrm{N}})^{-1}}\right)^{\mathbf{n}} \cdot \left(F^{\circ\mathbf{1}^{\mathrm{K}}}\right)^{-\boldsymbol{\rho}} \stackrel{?}{=} 1_{\mathbb{G}}. \tag{31}$$

The remaining constraints—that $\mathbf{n}$ is a binary vector of Hamming weight K and that $\boldsymbol{\rho} = (\boldsymbol{\alpha}+\gamma\cdot\mathbf{1}^{\mathrm{K}})^{-1}$—are enforced by our ZK protocol.

We now apply the KKDL-Embedding technique to the equalities (27)–(31) simultaneously. The verifier samples a random challenge u and verifies that:

$$\text{LHS of (27)} + \cdots + u^4 \cdot \text{LHS of (31)} \stackrel{?}{=} 1_{\mathbb{G}}. \tag{32}$$

This combined equation is then embedded into the commitment C via KKDL-Embedding through a change of generators. Concretely, the verifier samples an additional challenge e and applies a change of generators:

$$\widehat{\mathbf{G}} = \mathbf{G} \circ \left(\mathbf{pk}^{\circ\mathbf{v}^{\mathrm{N}}} \circ \mathbf{ct}^{\circ(u\cdot\mathbf{v}^{\mathrm{N}}+u^2\mathbf{1}^{\mathrm{N}})} \circ F^{\circ(u^4(\mathbf{v}^{\mathrm{N}}+\gamma\cdot\mathbf{1}^{\mathrm{N}})^{-1})}\right)^{e}, \tag{33}$$

$$\widehat{\mathbf{A}} = \mathbf{A} \circ \mathbf{tg}^{eu^3}, \qquad \widehat{\mathbf{B}} = \mathbf{B} \circ F^{\circ(-eu^4\mathbf{1}^{\mathrm{K}})}, \tag{34}$$

$$\widehat{H}_0 = H_0(GQ^{u^3})^{-e}, \qquad \widehat{H}_1 = H_1(W^u)^{-e}, \tag{35}$$

$$\widehat{H}_2 = H_2(E^u)^{-e}, \qquad \widehat{H}_3 = H_3(E^{u^2})^{-e}, \tag{36}$$

$$\widehat{\mathbf{H}}_{4:7} = \mathbf{H}_{4:7},\ C' = C \cdot \left(\prod_{i=0}^{\mathrm{M}-1} \mathsf{rct}_i\right)^{eu^2},\ \widehat{U} = U \tag{37}$$

Applying KKDL-Embedding to (32) for the commitment C yields Equations (33)–(37).

5.3 Applying Our ZK Protocol

The final stage of the protocol is devoted to enforcing the remaining field-level constraints captured by $\mathbf{R}_{\text{inner}}$. Specifically, the prover must convince the verifier that $\mathbf{n}$ is a binary vector of Hamming weight K, and that $(\boldsymbol{\alpha}+\gamma\cdot\mathbf{1}^{\mathrm{K}})\circ\boldsymbol{\rho} = \mathbf{1}^{\mathrm{K}}$. After this step, all constraints established so far are enforced.

Theorem 3. *UniBit constitutes an argument of knowledge for* $\mathbf{R}_{\text{RingCT}}$ *with statistical completeness, CWEE, and sHVZK.*

The formal proof is deferred to the full version of the paper.

6 Experimental Evaluation

We implement UniBit, UniBit$^+$, UniBitRange, Omniring*, and BulletCT in Rust using the Arkworks [3] framework over the Ristretto255 curve, with comparable engineering effort across all schemes. All experiments are conducted on a MacBook M1 Pro using a single thread.

Figure 7 reports the signing time and verification time. Figure 8 reports signature size. In our experiment, we vary the ring size N from 32 to 4096, the number of inputs K from 1 to 16, and set the number of outputs $\mathrm{M} \in \{2, 16\}$. Throughout all experiments, we fix the range-constraint bit length to $\beta = 64$.

Figure 9 further reports the signature size of UniBitRange under a small-K special-case optimization. Specifically, when K is small, we instantiate UniBitRange with a classic 3-move Σ protocol instead of the amortized compressed Σ protocol, which yields smaller signatures. In this experiment, we again vary N from 32 to 4096, consider $\mathrm{K} \in \{1, 4\}$, and $\mathrm{M} \in \{2, 64\}$.

Tables 3 and 4 summarize the concrete signing time, verification time, signature size, and end-to-end transaction size. These tables cover $\mathrm{M} \in \{2, 16\}$, $\mathrm{K} \in \{1, 2, 4, 8, 16\}$, and $\mathrm{N} \in \{256, 1024\}$.

6.1 Implementation Details

Instantiation of range-proof. Among the five schemes implemented in our evaluation, only UniBit and UniBit$^+$ have modular and interchangeable range-proof components (among the schemes in the theoretical comparison of Table 2, RingCT-3.0 [30] also adopts a separate range-proof design). For *ease of implementation*, our runtime benchmarks instantiate UniBit and UniBit$^+$ with our novel range-proof construction. These two schemes can also be instantiated with other compatible range-proof systems. Compared with our novel range-proof, Bulletproof^{++} achieves better proving time, verification time, and proof size, while also retaining a transparent setup in the pairing-free discrete-logarithm setting. Accordingly, Bulletproof^{++} is the more preferable instantiation for the range-proof component of UniBit and UniBit$^+$. Thus, for UniBit and UniBit+, the reported runtime and size correspond to two compatible instantiations: runtime is measured using our implemented range-proof, while size is reported for the preferred Bulletproof^{++} instantiation.

For UniBitRange, Omniring*, and BulletCT, the range constraint is *not* a swappable component. Instead, the "range-proof" is integrated into the main constraint system, and therefore the techniques used for handling range constraints must be compatible with the underlying construction. For example, Omniring* and BulletCT are fundamentally IPA-based designs, and thus they incorporate Bulletproof's IPA-based range-proof techniques directly into the overall constraint system. Similarly, UniBitRange is built upon the WNILA framework, and therefore integrates our novel norm-based range-proof construction into the protocol in an analogous manner. Consequently, one *cannot* simply instantiate UniBitRange with Bulletproof, nor replace the range-proof components of Omniring* or BulletCT with our novel range-proof scheme. Moreover, Bulletproof^{++}'s technique is *not* compatible with any of these constructions for constraint-level integration.

Optimization for BulletCT. To ensure a fully fair comparison, we apply an optimization to BulletCT's final IPA that

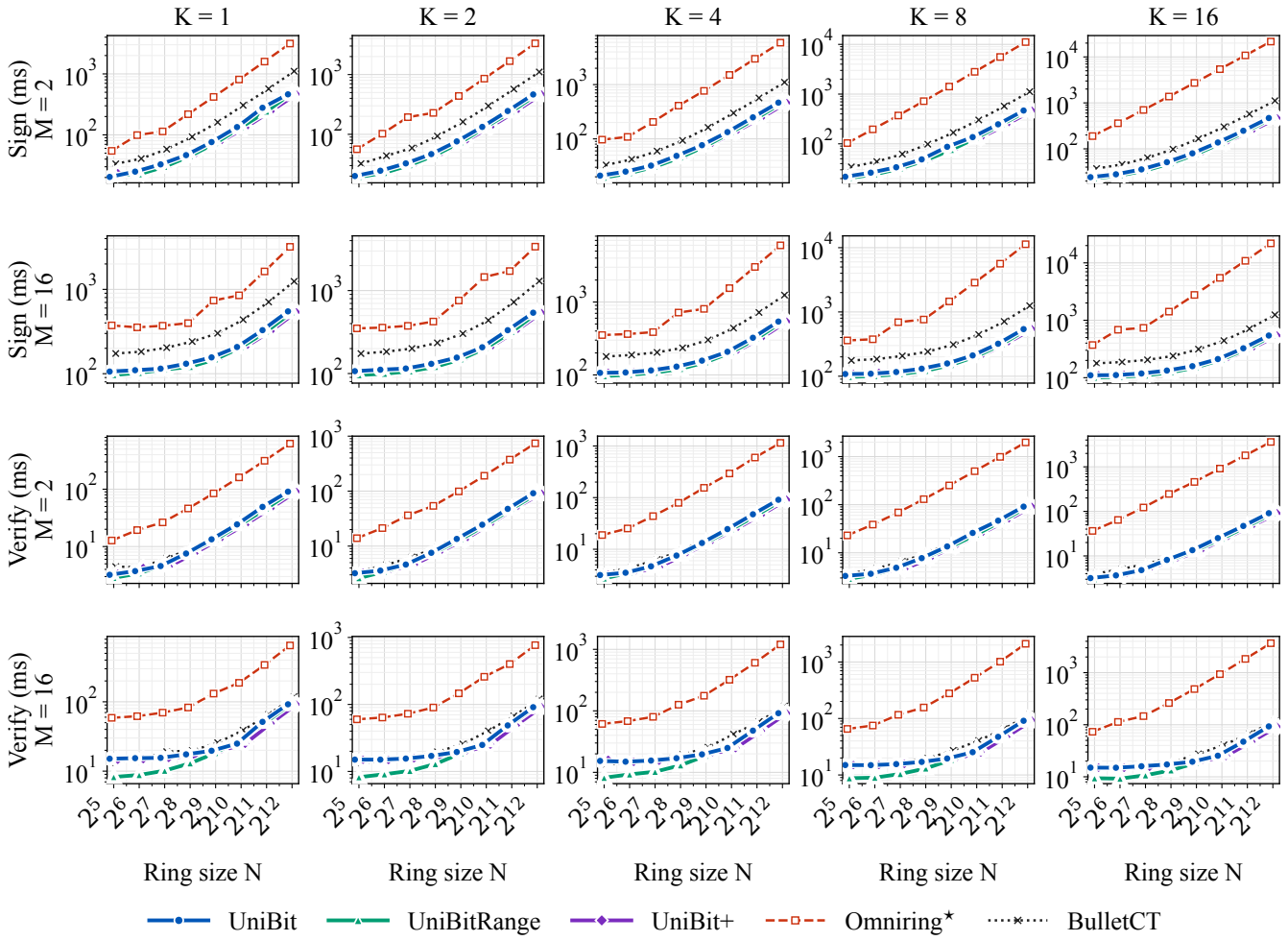

Figure 7: Signing and verification time for our benchmarks. Rows 1, 2 report signing time, rows 3, 4 report verification time.

is not described in the original BulletCT paper. BulletCT's IPA has size $2(N+\beta\cdot M+K)$, which is the sum of three components that are each padded to a power of two. This structure allows BulletCT's IPA to be instantiated as a three-fold IPA, analogous to our δ-fold WNILA, by simultaneously folding the $2N$, $2\beta\cdot M$, and $2K$ components. This optimization eliminates padding to the next power of two, which would otherwise introduce substantial overhead for both the signer and the verifier.

6.2 Signing Time

As shown in Figure 7 and Table 3, UniBit, UniBit$^+$, and UniBitRange consistently outperform Omniring* and BulletCT in signing time across all evaluated parameter settings. For $K=1, M=2$, UniBit requires approximately 20ms at $N=32$ and 465ms at $N=4096$, whereas Omniring* requires approximately 55ms and 3187ms, and BulletCT requires approximately 31ms and 1094ms, respectively. When $K=16, M=2$ and $N=4096$, our schemes still complete signing in approximately 469ms, while BulletCT requires around 1115ms and Omniring* requires approximately 21818ms. For $M=16, K=16, N=4096$, UniBit's signing time is 542ms while BulletCT's signing time is 1246ms. Overall, our schemes achieve a signing speedup exceeding $40\times$ in the largest evaluated signing regime, and a speedup of 1.6–$2.3\times$ against BulletCT.

These results are consistent with the theoretical analysis. Compared with Omniring*, WNILA reduces the witness size by a factor of two, while the integration of K-Weight, K-Link, and KKDL further reduces the witness length from $O(NK)$ to $O(N+K)$. The advantage of our schemes becomes increasingly significant as K grows. Compared with BulletCT, the advantage comes from WNILA.

Table 3: Sign/verify in milliseconds with $\beta = 64$. Darker gray cells indicate larger speedups over BulletCT.

Parameters			**UniBit**		**UniBit+**		**UniBitRange**		BulletCT		Omniring*	
M	N	K	Sign	Verify	Sign	Verify	Sign	Verify	Sign	Verify	Sign	Verify
2	256	1	46.507	7.550	47.371	7.584	45.020	8.017	92.966	9.454	218.422	46.117
16	256	1	131.337	17.469	128.394	16.871	120.688	12.980	240.577	19.812	397.727	82.823
2	256	2	46.675	7.551	47.059	7.815	45.001	7.817	93.374	9.625	228.285	53.935
16	256	2	130.875	17.040	128.950	16.817	120.552	12.913	234.488	19.657	423.350	89.985
2	256	4	48.548	7.547	48.231	7.932	45.649	7.873	93.601	9.741	409.512	79.218
16	256	4	129.949	16.985	129.355	16.915	121.226	13.025	235.026	19.462	716.660	125.885
2	256	8	48.605	7.742	49.578	7.704	46.774	7.984	98.239	10.096	720.459	130.192
16	256	8	130.873	17.025	130.361	16.846	121.532	13.051	241.304	19.812	761.875	156.540
2	256	16	51.230	8.145	51.900	8.256	48.709	8.183	98.039	9.941	1374.627	246.490
16	256	16	133.160	17.246	132.253	16.831	124.091	13.313	238.953	19.281	1415.126	261.081
2	1024	1	134.143	24.323	131.408	24.004	133.358	24.683	305.658	30.524	809.159	160.769
16	1024	1	207.194	24.935	206.796	24.813	205.764	29.365	436.712	38.948	849.033	187.781
2	1024	2	133.076	24.474	132.967	25.387	134.611	24.565	298.601	29.442	854.773	190.399
16	1024	2	207.290	24.864	206.811	25.099	204.383	29.307	437.997	40.368	1458.080	258.919
2	1024	4	133.270	24.575	134.533	24.439	132.680	24.419	299.232	29.474	1505.792	293.513
16	1024	4	207.746	24.903	207.551	24.784	205.030	29.266	438.706	41.748	1548.636	320.414
2	1024	8	136.316	25.906	133.882	24.833	132.756	24.279	302.006	29.977	2808.202	497.428
16	1024	8	210.494	25.318	208.392	25.000	205.228	29.446	446.050	40.815	2858.075	526.921
2	1024	16	137.454	25.155	139.323	25.248	134.417	24.562	301.918	29.931	5416.950	913.869
16	1024	16	211.108	25.397	211.297	25.142	207.953	29.757	442.052	41.074	5488.427	934.643

6.3 Verification Time

As shown in Figure 7 and Table 3, UniBit, UniBit$^+$, and UniBitRange significantly outperform Omniring* in verification time across all evaluated parameter settings, while achieving more moderate improvements over BulletCT.

For $K = 1, M = 2$, UniBit requires approximately 3ms to verify a signature at $N = 32$ and 91ms at $N = 4096$, whereas Omniring* requires approximately 13ms and 633ms, respectively. At larger parameter settings, the gap becomes even more pronounced: when $M = 2, K = 16$ and $N = 4096$, UniBit requires approximately 94ms for verification, while Omniring* requires approximately 3683ms, yielding a speedup of roughly $40\times$.

Compared with BulletCT, when $M = 16$, $N \in \{32, 64\}$ and $K = 16$, UniBitRange achieves the fastest verification time, requiring only 8–9ms, while UniBit, UniBit$^+$, and BulletCT all require approximately 14–15ms. When $N = 1024$, $K = 16$, and $M = 16$, UniBit and UniBit$^+$ require approximately 25ms for verification, UniBitRange requires approximately 29ms, and BulletCT requires approximately 41ms. When $M = 16, K = 16, N = 4096$, UniBit and UniBit$^+$ take 91ms, UniBitRange takes 96ms, while BulletCT takes 119ms. Overall, the relative verification cost varies with the construction and parameters. Across the evaluated settings, we observe verification speedups of roughly 1.1–$1.65\times$ over BulletCT.

These results are consistent with the theoretical analysis. Compared with Omniring*, the reduction in witness size achieved by combining WNILA with the K-Weight and K-Link techniques directly translates into a substantial reduction in the verifier's final MSM size, thereby yielding significant verification speedups. Compared with BulletCT, WNILA reduces the size of the verifier's final MSM by $N + \beta \cdot M$. In addition, the range-proof verification MSMs in UniBit and UniBit$^+$ can be batched together with other MSM computations due to the shared generators.

6.4 Signature Size

We calculate signature sizes from the protocol specifications and present them in Figure 8 and Table 4. For UniBit and UniBit$^+$, we assume that range-proofs are instantiated using Bulletproof^{++} [12] and adopt the proof-size figures reported therein. UniBit is slightly larger than Omniring* at small ring sizes—for example, 1.781 KB versus 1.312 KB at $N = 32$, but this gap gradually narrows as N increases. UniBit$^+$ further reduces the signature size of UniBit by 0.125 KB, and UniBitRange achieves smaller signatures than Omniring* once $N \geq 2^8$. UniBitRange is smaller because it handles range constraints in a non-black-box manner and therefore avoids the separate-range-proof overhead. Across the representative settings in Table 4, BulletCT remains somewhat smaller: the gap is small for UniBitRange and larger for UniBit/UniBit+ due to the separate range-proof component. For example, when

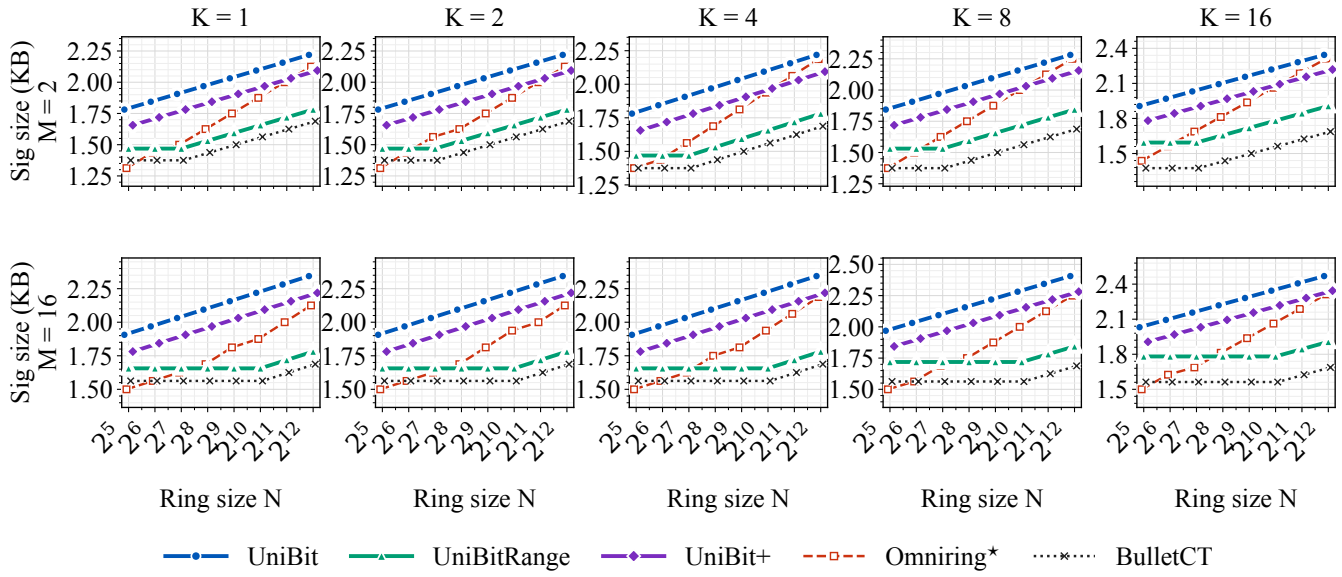

Figure 8: Signature size for our benchmarks. For UniBit and UniBit$^+$, we assume they are instantiated with Bulletproof^{++} [12].

Table 4: Signature/transaction sizes in KB with $\beta = 64$. Darker gray cells indicate smaller overhead relative to BulletCT.

Parameters			**UniBit**		**UniBit+**		**UniBitRange**		BulletCT		Omniring⋆	
M	N	K	Sg	Tx	Sg	Tx	Sg	Tx	Sg	Tx	Sg	Tx
2	256	1	1.969	3.125	1.844	3.000	1.531	2.688	1.438	2.594	1.625	2.781
16	256	1	2.094	4.125	1.969	4.000	1.656	3.688	1.562	3.594	1.688	3.719
2	256	2	1.969	3.156	1.844	3.031	1.531	2.719	1.438	2.625	1.625	2.812
16	256	2	2.094	4.156	1.969	4.031	1.656	3.719	1.562	3.625	1.688	3.750
2	256	4	1.969	3.219	1.844	3.094	1.531	2.781	1.438	2.688	1.688	2.938
16	256	4	2.094	4.219	1.969	4.094	1.656	3.781	1.562	3.688	1.750	3.875
2	256	8	2.031	3.406	1.906	3.281	1.594	2.969	1.438	2.812	1.750	3.125
16	256	8	2.156	4.406	2.031	4.281	1.719	3.969	1.562	3.812	1.750	4.000
2	256	16	2.094	3.719	1.969	3.594	1.656	3.281	1.438	3.062	1.812	3.438
16	256	16	2.219	4.719	2.094	4.594	1.781	4.281	1.562	4.062	1.812	4.312
2	1024	1	2.094	6.250	1.969	6.125	1.656	5.812	1.562	5.719	1.875	6.031
16	1024	1	2.219	7.250	2.094	7.125	1.656	6.688	1.562	6.594	1.875	6.906
2	1024	2	2.094	6.281	1.969	6.156	1.656	5.844	1.562	5.750	1.875	6.062
16	1024	2	2.219	7.281	2.094	7.156	1.656	6.719	1.562	6.625	1.938	7.000
2	1024	4	2.094	6.344	1.969	6.219	1.656	5.906	1.562	5.812	1.938	6.188
16	1024	4	2.219	7.344	2.094	7.219	1.656	6.781	1.562	6.688	1.938	7.062
2	1024	8	2.156	6.531	2.031	6.406	1.719	6.094	1.562	5.938	2.000	6.375
16	1024	8	2.281	7.531	2.156	7.406	1.719	6.969	1.562	6.812	2.000	7.250
2	1024	16	2.219	6.844	2.094	6.719	1.781	6.406	1.562	6.188	2.062	6.688
16	1024	16	2.344	7.844	2.219	7.719	1.781	7.281	1.562	7.062	2.062	7.562

$M = 16, N = 1024, K = 4$, BulletCT is 0.094KB smaller than UniBitRange, 0.532KB smaller than UniBit$^+$ and 0.657KB smaller than UniBit.

Computation and communication trade-off. We next examine why our schemes have larger signature sizes than BulletCT. The compression pivot itself is not the main source of overhead: WNILA requires $2\log(\max(N, \beta \cdot M))$ group elements and 4 scalars, while the 3-fold IPA requires

$2\log(\max(N, \beta \cdot M))$ group elements and 6 scalars. Thus, the two designs have comparable compression-pivot sizes. The additional communication cost in our schemes mainly comes from two components: the separate range-proof design in UniBit and UniBit$^+$, and the amortized compressed-Σ-protocol used by all three of our schemes.

The separate range-proof design is a deliberate design choice rather than an inherent limitation. It causes UniBit and UniBit$^+$ to include a *standalone* range-proof, whereas UniBitRange integrates the range constraints directly and therefore achieves shorter signatures. This modularity, however, brings practical advantages: it simplifies implementation, allows UniBit and UniBit$^+$ to adopt state-of-the-art compatible range-proof systems, and makes the range-proof MSMs batchable with other MSMs in the RingCT signature. Such batching is not available to UniBitRange, BulletCT, or Omniring*.

The overhead is the amortized compressed Σ-protocol [4], which proves the well-formedness of the commitments V and P. Omniring* requires a similar compressed Σ protocol as well. For small K, this component can be replaced by a classic 3-move Σ-protocol, which yields smaller signatures. The amortized compressed Σ protocol has a relatively large constant overhead, namely 5 group elements and 9 scalars. UniBit and UniBitRange are amenable to this small-K optimization. As shown in Figure 9, when $K \leq 4$, this optimization enables UniBitRange to achieve smaller signatures than BulletCT. This will not hurt the signing or verification time since the classic 3-move Σ protocol is computationally faster than amortized compressed-Σ protocol. We note that the same optimization is less effective for UniBit$^+$, whose construction inherently requires additional randomization scalars per commitment for zero-knowledge. Moreover, Omniring* cannot adopt this optimization because its Σ-protocol's witness length is $O(N)$.

6.5 Transaction Size

The "Tx" columns of Table 4 report the transaction size of UniBit, UniBit$^+$, UniBitRange, BulletCT and Omniring*. We use the exact same transaction size calculation approach as BulletCT [28]. For each member in the decoy ring, we count a 4-byte pointer to the corresponding ring account. Each tag is counted as a 32-byte compressed group element. Each output consists of one output coin commitment and one target account public key, i.e., two compressed group elements. The transaction-size difference among UniBit, UniBit+, UniBitRange, BulletCT, and Omniring* is determined only by the RingCT signature size. We note that ZGSX23's Any-out-of-N RingCT signature is functionally different from our schemes and is not suitable for an apples-to-apples comparison. Moreover, as discussed in BulletCT [28], ZGSX23's transaction size is much larger than that of any scheme in Table 4, since each transaction must include N tags.

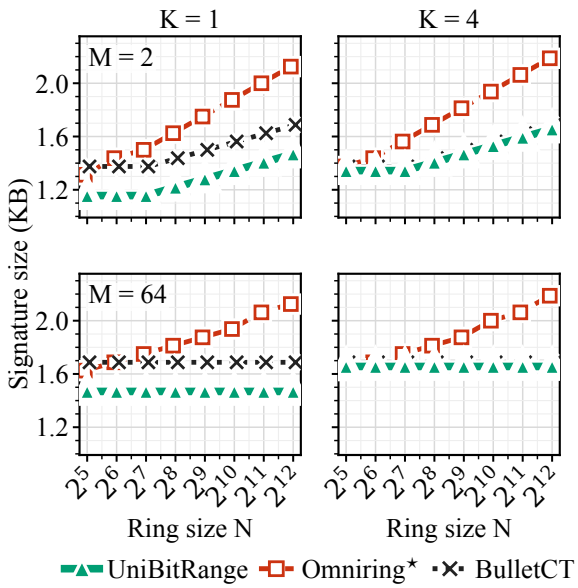

Figure 9: Signature-size comparison when using classic 3-move Σ protocol for UniBitRange instead of the amortized compressed Σ-protocol, for $K \leq 4$. We set $M \in \{2, 64\}$ and $\beta = 64$, while varying N from 32 to 4096.

7 Conclusion

We identify pervasive bottlenecks in prior Bulletproof-based RingCT signature schemes and present new techniques that address these inefficiencies. Together, these techniques lead to a state-of-the-art design for Bulletproof-based RingCT signature schemes. Our results represent a concrete step forward for more scalable private payment technologies.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under grant 62572038, 62303037, 62272464 and U24B20144, the Key R&D Program of Zhejiang Province (No. 2025C01084), the Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ26F030002, Beijing-Tianjin-Hebei natural science foundation cooperation project 25JJJJC0033, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China(No. 202530187).

Ethical Considerations

Stakeholder Identification and Impacts. We organize our ethical analysis around a stakeholder-based perspective. In

particular, we distinguish between direct stakeholders, who are involved in the design, implementation, or deployment of our schemes, and indirect stakeholders, who may be affected by systems built upon them. Direct stakeholders include the authors and organizations that integrate our schemes into blockchain platforms, distributed systems, or privacy-preserving applications. Indirect stakeholders include end users whose assets or data depend on such systems, regulatory bodies and auditors responsible for oversight, as well as adversaries who may seek to exploit or undermine these technologies.

Research Team. This work does not involve human subjects, physical experimentation, or the collection of real-world personal data. We ensure that all intellectual contributions are properly acknowledged and that authorship accurately reflects substantial technical involvement.

Blockchain and Privacy Technology Companies. Organizations adopting RingCT or zero-knowledge proof techniques may benefit from the improved efficiency offered by our schemes. At the same time, integration may require non-trivial engineering effort, incur additional costs for security audits, or exhibit performance trade-offs under certain parameter regimes. To support informed adoption, we report concrete performance measurements, including signing time, verification time, and signature size, and explicitly document known limitations, such as increased signature size when the number of outputs is large.

Cryptography Community and Society.

Beneficence. We introduce new compression techniques, zero-knowledge proof constructions, and embedding methods for RingCT signatures that enable more efficient private transactions in UTXO-based blockchains. Our contributions aim to strengthen confidentiality and integrity guarantees while avoiding access to users' raw transaction data, thereby aligning with established ethical principles for data security and privacy.

Respect for Persons. We carefully reference and credit prior work to ensure accurate attribution and to respect the contributions of the broader research community.

Justice. Our results are purely mathematical and algorithmic in nature and do not rely on proprietary components or specialized infrastructure. By providing detailed descriptions and analysis, we aim to keep the core ideas accessible to researchers and practitioners worldwide and to reduce barriers to entry in this research area.

Respect for Law and Public Interest. Our research adheres to applicable academic and cryptographic research norms and policies. By advancing the efficiency and auditability of RingCT signatures, this work serves the public interest by supporting the development of more trustworthy and robust digital financial infrastructure.

Publication Impact. Our schemes may shape future research directions by enabling more efficient RingCT signatures and zero-knowledge proof systems. To mitigate potential misuse or misunderstanding, we explicitly discuss key design trade-offs, including the effects of the parameters N, K and M, to guide responsible interpretation and deployment.

Potential Harms Analysis. As with many privacy-enhancing technologies, our protocols could be misused by illicit actors to conceal unlawful activities, such as obscuring financial flows within privacy-preserving systems. Additionally, premature or improperly reviewed implementations may pose risks to honest users, since cryptographic flaws can result in loss of funds, privacy violations, or service disruptions. We therefore emphasize responsible deployment within the bounds of applicable law and regulation and encourage implementers to combine our schemes with appropriate governance, monitoring, and compliance mechanisms. We also clearly state the underlying security assumptions to reduce the risk of unsafe deployment.

Decision to Conduct and Publish. This research was motivated by the goal of improving the efficiency of existing RingCT signatures and zero-knowledge proof constructions. The project was approved internally after evaluating its technical merit and consistency with our research objectives. We considered the dual-use nature of more efficient privacy mechanisms and concluded that the anticipated benefits to privacy-preserving systems outweigh the risks of misuse when appropriate legal and technical safeguards are applied.

The decision to publish followed extensive internal validation, including proof checking, peer review among the authors, and empirical benchmarking of our implementations. We believe the results offer sufficient novelty, correctness, and practical value to justify dissemination to the cryptographic community.

Although residual dual-use risks cannot be entirely eliminated, we did not identify ethical concerns significant enough to preclude publication. We believe that open dissemination promotes transparency, accountability, and scientific progress, consistent with established norms in cryptography research.

Open Science

Our code is available at: https://doi.org/10.5281/zenodo.20376707

References

[1] Sina Aeeneh, João Otávio Chervinski, Jiangshan Yu, and Nikola Zlatanov. New attacks on the untraceability of transactions in cryptonote-style blockchains. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5, 2021. URL: https://ieeexplore.ieee.org/document/9461130/, doi:10.1109/ICBC51069.2021.9461130.

[2] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user

privacy in bitcoin. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 34–51, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-642-39884-1_4.

[3] arkworks contributors. `arkworks` zksnark ecosystem, 2022. URL: https://arkworks.rs.

[4] Thomas Attema and Ronald Cramer. Compressed sigma-protocol theory and practical application to plug and play secure algorithmics. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 513–543, Cham, 2020. Springer International Publishing. URL: https://dl.acm.org/doi/10.1007/978-981-96-9095-4_9.

[5] Danai Balla and Pyrros Chaidos. On the (strong) linkability of linkable ring signatures. Cryptology ePrint Archive, Paper 2025/1375, 2025. URL: https://eprint.iacr.org/2025/1375.

[6] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better — how to make bitcoin a better currency. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security*, pages 399–414, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-642-32946-3_29.

[7] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, page 967–980, New York, NY, USA, 2013. Association for Computing Machinery. URL: https://dl.acm.org/doi/10.1145/2508859.2516734.

[8] Benedikt Bunz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, Los Alamitos, CA, USA, May 2018. IEEE Computer Society. URL: https://ieeexplore.ieee.org/document/8418611.

[9] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 78–96, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. URL: https://doi.org/10.1007/11818175_5.

[10] Alishah Chator and Matthew Green. How to squeeze a crowd: Reducing bandwidth in mixing cryptocurrencies. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*, pages 40–49, 2018. URL: https://ieeexplore.ieee.org/document/8406559, doi:10.1109/EuroSPW.2018.00012.

[11] Heewon Chung, Kyoohyung Han, Chanyang Ju, Myungsun Kim, and Jae Hong Seo. Bulletproofs+: Shorter proofs for privacy-enhanced distributed ledger. Cryptology ePrint Archive, Paper 2020/735, 2020. URL: https://eprint.iacr.org/2020/735.

[12] Liam Eagen, Sanket Kanjalkar, Tim Ruffing, and Jonas Nick. Bulletproofs++: Next generation confidential transactions via reciprocal set membership arguments. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 249–279, Cham, 2024. Springer Nature Switzerland. URL: https://link.springer.com/chapter/10.1007/978-3-031-58740-5_9.

[13] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg. URL: https://link.springer.com/chapter/10.1007/3-540-47721-7_12.

[14] Brandon Goodell, Sarang Noether, and Arthur Blue. Concise linkable ring signatures and forgery against adversarial keys. Cryptology ePrint Archive, Paper 2019/654, 2019. URL: https://eprint.iacr.org/2019/654.

[15] Jens Groth and Yuval Ishai. Sub-linear zero-knowledge argument for correctness of a shuffle. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 379–396, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-540-78967-3_22.

[16] Ulrich Haböck. Multivariate lookups based on logarithmic derivatives. Cryptology ePrint Archive, Paper 2022/1530, 2022. URL: https://eprint.iacr.org/2022/1530.

[17] Nada Hammad and Friedhelm Victor. Monero traceability heuristics: Wallet application bugs and the mordinal-p2pool perspective. In *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 540–548, 2024. URL: https://ieeexplore.ieee.org/document/10634447.

[18] Xiangyu Hui and Sid Chi-Kin Chau. Llring: Logarithmic linkable ring signatures with transparent setup. In Joaquin Garcia-Alfaro, Rafa Kozik, Micha Chora, and Sokratis Katsikas, editors, *Computer Security – ESORICS 2024*, pages 299–319, Cham, 2024. Springer Nature Switzerland. URL: https://link.springer.com/chapter/10.1007/978-3-031-70896-1_15.

[19] Philip Koshy, Diana Koshy, and Patrick McDaniel. An analysis of anonymity in bitcoin using p2p network traffic. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 469–485, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-662-45472-5_30.

[20] Russell W. F. Lai, Viktoria Ronge, Tim Ruffing, Dominique Schröder, Sri Aravinda Krishnan Thyagarajan, and Jiafan Wang. Omniring: Scaling private payments without trusted setup. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 31–48, New York, NY, USA, 2019. Association for Computing Machinery. URL: https://dl.acm.org/doi/10.1145/3319535.3345655.

[21] Jonathan Lee. Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography*, pages 1–34, Cham, 2021. Springer International Publishing. URL: https://link.springer.com/chapter/10.1007/978-3-030-90453-1_1.

[22] Li Lin, Tian Qiu, Xin Wang, Hailong Wang, Changzheng Wei, Ying Yan, Wei Wang, and Wenbiao Zhao. Zero-knowledge protocol for knowledge of known discrete logarithms: Applications to ring confidential transactions and anonymous zether. Cryptology ePrint Archive, Paper 2025/690, 2025. URL: https://eprint.iacr.org/2025/690.

[23] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 171–189, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. URL: https://link.springer.com/chapter/10.1007/3-540-44647-8_10.

[24] Monero Research Lab. *Zero to Monero: Second Edition*. Monero Research Lab, 2020. URL: https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf.

[25] Shen Noether, Adam Mackenzie, and the Monero Research Lab. Ring confidential transactions. *Ledger*, 1:1–18, Dec. 2016. URL: https://www.getmonero.org/es/resources/research-lab/pubs/MRL-0005.pdf.

[26] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual International Cryptology Conference*, 1991. URL: https://api.semanticscholar.org/CorpusID:13447431.

[27] The-Anh Ta, Xiangyu Hui, and Sid Chi-Kin Chau. Ring referral: Efficient publicly verifiable ad hoc credential scheme with issuer and strong user anonymity for decentralized identity and more. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 184–202, Los Alamitos, CA, USA, May 2025. IEEE Computer Society. URL: https://ieeexplore.ieee.org/document/11023338.

[28] Nan Wang, Qianhui Wang, Dongxi Liu, Muhammed F. Esgin, and Alsharif Abuadbba. Bulletct: towards more scalable ring confidential transactions with transparent setup. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA, 2025. USENIX Association. URL: https://dl.acm.org/doi/10.5555/3766078.3766250.

[29] Xingyu Yang, Lei Xu, and Liehuang Zhu. Deanonymizing monero: A maximum weighted matching-based approach. *IEEE Transactions on Information Forensics and Security*, 20:4726–4738, 2025. URL: https://ieeexplore.ieee.org/document/10963663, doi:10.1109/TIFS.2025.3560193.

[30] Tsz Hon Yuen, Shi-Feng Sun, Joseph K. Liu, Man Ho Au, Muhammed F. Esgin, Qingzhao Zhang, and Dawu Gu. Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security*, pages 464–483, Cham, 2020. Springer International Publishing. URL: https://link.springer.com/chapter/10.1007/978-3-030-51280-4_25.

[31] Tianyu Zheng, Shang Gao, Yubo Song, and Bin Xiao. Leaking arbitrarily many secrets: Any-out-of-many proofs and applications to ringct protocols. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2533–2550, Los Alamitos, CA, USA, May 2023. IEEE Computer Society. URL: https://ieeexplore.ieee.org/document/10179292.

A Related Work

Bulletproof techniques. Bulletproof [8] is a general-purpose proof system for range-proofs and circuit satisfiability. At a high level, its proof technique follows a two-stage paradigm. In the first stage, a cryptographic compiler reduces a general constraint system to a collection of inner-product constraints while preserving zero-knowledge. In the second stage, these constraints are discharged using an inner-product argument with logarithmic communication complexity.

Two notable follow-up works along this line are Bulletproof^{+} [11] and Bulletproof^{++} [12]. Bulletproof^{+} refines the original inner-product argument to directly sup-

port weighted inner products and integrates zero-knowledge randomization into this argument. As a result, Bulletproof^{+} adopts a more streamlined protocol design and no longer exhibits the explicit two-stage structure of Bulletproof, while achieving improved concrete efficiency.

Bulletproof^{++} revisits the original two-stage paradigm with several substantial modifications. In particular, it replaces the inner-product argument with a weighted norm linear argument (WNLA) and introduces a more expressive compiler that reduces constraints in its circuit model directly to WNLA instances. Moreover, Bulletproof^{++} proposes a randomization technique that leverages the linear component of WNLA to blind the witness, providing an efficient mechanism for achieving zero-knowledge with reduced communication overhead. These techniques enable the circuit model of Bulletproof^{++} to be instantiated as a highly efficient range-proof system. Our constructions build on and further extend these ideas.

Bulletproof-based RingCT signatures. Bulletproof-based RingCT signatures can be viewed as signatures of knowledge for a relation $\mathbf{R}_{\text{RingCT}}$. Two concurrent works, RingCT 3.0 [30] and Omniring [20], were the first to leverage Bulletproof techniques to construct RingCT signatures. RingCT 3.0 represents the selection of real inputs using a committed bit vector $\mathbf{n} \in \{0,1\}^{\text{KN}}$ that encodes the membership information of the input ring. The corresponding public keys and linking tags are embedded into this commitment, and the binary constraints on $\mathbf{n}$ are enforced using the inner-product argument from Bulletproof. Lin et al. [22] later examined this construction in the context of the KKDL issue presenting a concrete cryptographic attack.

Omniring [20] proposes an alternative approach that explicitly addresses the KKDL issue. Its core idea is to let the verifier freshly sample the generators, enabling a security analysis based on the interactive discrete-logarithm (iDL) assumption. In its original presentation, the security proof of Omniring employs the Schwartz–Zippel lemma to argue that certain vector equalities hold with high probability. This argument requires that the relevant vectors be fixed prior to sampling the random challenge. However, in Omniring these vectors depend on the challenge, necessitating additional care in the analysis. Lin et al. [22] resolve this issue by refining the interaction order and incorporating an inline compressed Σ-protocol, ensuring that the witness elements remain independent of the challenge when applying the Schwartz–Zippel lemma. Combining this refinement with the original Omniring construction yields a clean and effective technique, which we refer to as the KKDL-Embedding technique. This technique enables direct proofs of exponentiation relations of the form $\mathbf{A}^{\mathbf{w}} \cdot \mathbf{B}^{\mathbf{c}} = 1_{\mathbb{G}}$, thereby extending the class of relations supported by Bulletproof-style constructions.

ZGSX23 [31] introduces the K-Weight technique for enforcing the K-out-of-N constraint. Instead of proving K independent 1-out-of-N statements, the construction establishes the existence of a random linear combination of secret keys, reducing the witness size from 2KN to 2N. This technique is compatible with the KKDL-Embedding framework, and our treatment of the K-out-of-N constraint naturally incorporates the core idea of ZGSX23 within the WNILA framework. BulletCT [28] further proposes the K-Link technique to enforce linkability by observing that it suffices to prove a permutation relation between public keys and their corresponding linking tags. This approach is compatible with our framework, as the resulting constraints can be modularly captured using WNILA together with the KKDL-Embedding technique.

B Our Novel Range Proof Scheme

We sketch our novel range proof scheme, which is a zero-knowledge argument of knowledge for the following relation:

$$\mathbf{R}_{\text{RP}} = \left\{ (\mathsf{rct}_i)_{i=0}^{\text{M}-1}; (a_i, r_i)_{i=0}^{\text{M}-1} \;\middle|\; \begin{array}{l} \forall i \in [\text{M}] : \mathsf{rct}_i = W^{a_i} E^{r_i} \\ \forall i \in [\text{M}] : 0 \le a_i \le 2^{\beta} - 1 \end{array} \right\}$$

Let $\mathbf{n} \in \{0,1\}^{\beta \cdot M}$ denote the binary representation of $\mathbf{a}$. We want to prove that

$$\mathbf{n} \in \{0,1\}^{\beta \cdot \text{M}} \text{ and for all } i \in [\text{M}] : a_i = \sum_{j=0}^{\beta-1} n_{i \cdot \beta + j} \cdot 2^j$$

Let μ, κ be two random challenges. The above constraint can be compressed as

$$\|\mathbf{n} - \frac{1}{2} \cdot \mathbf{1}^{\beta \cdot \text{M}}\|_{\mu}^2 = \frac{1}{4} \cdot \sum_{i=0}^{\beta \cdot \text{M}-1} \mu^{i+1}$$

and

$$\langle \mathbf{n}, \sum_{i=0}^{\text{M}-1} \kappa^{i+1} \cdot (\mathbf{0}^{(i \cdot \beta)} || \mathbf{2}^{\beta} || \mathbf{0}^{(\text{M}-i-1) \cdot \beta}) \rangle = \sum_{i=0}^{\text{M}-1} \kappa^{i+1} \cdot a_i$$

This is captured through the following polynomial encoding:

$$\begin{aligned} \mathbf{n}(X) = \mathbf{n} - \frac{1}{2} \cdot \mathbf{1}^{\beta \cdot \text{M}} \\ + \mu^{-1} \cdot \boldsymbol{\mu}^{-\beta \cdot \text{M}} \circ \left(\sum_{i=0}^{\text{M}-1} \kappa^{i+1} \cdot (\mathbf{0}^{i \cdot \beta} \| \mathbf{2}^{\beta} \| \mathbf{0}^{(\text{M}-i-1) \cdot \beta}) \right) \\ + \mathbf{s}_G \cdot X \end{aligned}$$

Then it is clear that

$$\|\mathbf{n}(X)\|_{\mu}^2 = t_0 + t_1 \cdot X + t_2 \cdot X^2$$

where

$$t_0 = \frac{1}{4} \cdot \sum_{i=0}^{\beta \cdot \text{M}-1} \mu^{i+1} + 2 \cdot \sum_{i=0}^{\text{M}-1} \kappa^{i+1} \cdot a_i + \delta(\mu, \kappa)$$

for which $\delta(\mu, \kappa)$ is defined as:

$$\mathbf{u} = \sum_{i=0}^{\text{M}-1} \kappa^{i+1} \cdot (\mathbf{0}^{(i \cdot \beta)} || \mathbf{2}^{\beta} || \mathbf{0}^{(\text{M}-i-1) \cdot \beta})$$

$$\delta(\mu,\kappa) = -\sum_{i=0}^{\beta\cdot M-1} u_i + \mu^{-i-1}\cdot u_i^2.$$

From these equations, we can naturally derive the range-proof scheme following the techniques in Section 4. Concretely, the prover commits to the witness vector $\mathbf{n}$ in A and the blinding vector $\mathbf{s}$ in S, after which both parties homomorphically derive the commitment to $\mathbf{n}(x)$. The verifier can compute $\hat{t}_0$ directly, and a WNILA instance with trivial inner-product and linear constraints can then be used to compress the communication complexity to logarithmic size. We provide the formal details in the full version of the paper.

Theorem 4. *This range proof achieves completeness, sHVZK and CWEE.*

C Deriving UniBitRange from Minor Modifications of UniBit

In this section, we sketch the main idea of UniBitRange. At a high level, the construction is obtained by "concatenating" the polynomial encoding $\mathbf{n}_1(X)$ for the range constraints into the UniBit framework. Let $\mathbf{n} = \mathbf{n}_0||\mathbf{n}_1$, where $\mathbf{n}_0$ is the bit vector for K-out-of-N and $\mathbf{n}_1$ is the bit vector for range constraints. We can modify the polynomial $\mathbf{n}(X)$ to

$$\begin{aligned}\mathbf{n}(X) &= \mathbf{n} - \frac{1}{2}\cdot\mathbf{1}^{N+\beta\cdot M}\\ &+ \kappa\cdot\left(\mu_0^{-1}\boldsymbol{\mu}_0^{-N}||\left(\mu_1^{-1}\boldsymbol{\mu}_1^{-\beta\cdot M}\circ\mathbf{w}\right)\right)\\ &+ \mathbf{s}_G\cdot X\end{aligned} \tag{38}$$

where $\mathbf{w} = \sum_{i=0}^{M-1}\kappa^{i+1}\cdot(\mathbf{0}^{(i\cdot\beta)}||\mathbf{2}^{\beta}||\mathbf{0}^{(M-i-1)\cdot\beta})$. One can see that this is just a concatenation of the polynomial encoding $\mathbf{n}_0(X)$ from UniBit and the polynomial encoding of $\mathbf{n}_1(X)$ from the range proof, with increasing power of μ_0 and μ_1. The remaining polynomial encodings are the same as UniBit.

From this, one can see that the constant c in the final part of the inner protocol becomes

$$\begin{aligned}c = &\frac{1}{4}\cdot\left(\sum_{i=0}^{N-1}\mu_0^{i+1} + \sum_{i=0}^{\beta\cdot M-1}\mu_1^{i+1}\right) + 2\kappa\cdot K\\ &+\kappa^2\cdot\sum_{i=0}^{K-1}\eta^i + \delta(\mu,\kappa)\end{aligned}$$

$$\delta(\mu,\kappa) = -\sum_{i=0}^{N+\beta\cdot M-1} u_i + \mu^{-i-1}\cdot u_i^2.$$

where

$$\mathbf{u} = \kappa\cdot\mu_0^{-1}\boldsymbol{\mu}_0^{-N}||\left(\mu_1^{-1}\boldsymbol{\mu}_1^{-\beta\cdot M}\circ\mathbf{w}\right) \tag{39}$$

Given these observations, UniBit can be naturally extended to UniBitRange. Instead of invoking a separate range-proof scheme, the prover commits to the concatenated vector $\mathbf{n}_0||\mathbf{n}_1$ in A, while the remaining parts of the inner protocol follow analogously to UniBit. The verifier can still homomorphically derive the commitment to $\mathbf{n}(x)$ and compute $\hat{t}_0$ directly, while WNILA continues to serve as the compression pivot. We leave the formal details to the full version of the paper.

Theorem 5. *UniBitRange achieves statistical completeness, CWEE and sHVZK for* $\mathbf{R}_{\text{RingCT}}$.

D Deriving UniBit$^+$ From Minor Modifications of UniBit

UniBit$^+$ only modifies how to achieve zero-knowledge. It adapts Bulletproof++ [12]'s blinding techniques. Recall that in UniBit, polynomial encodings we are considering is

$$\begin{aligned}\mathbf{n}(X) &= \mathbf{n} - \frac{1}{2}\mathbf{1}^N - \kappa\mu^{-1}\mu^{-N} + \mathbf{s}_G\cdot X,\\ \mathbf{a}(X) &= \mathbf{a} + \mathbf{s}_A\cdot X,\\ \mathbf{b}(X) &= \kappa^2\left(\boldsymbol{\eta}^K\circ(\boldsymbol{b} + \boldsymbol{s}_B\cdot X)\right),\\ \mathbf{l}(X) &= \mathbf{l} + \mathbf{s}_H\cdot X.\end{aligned}$$

Then, we have that

$$\begin{aligned}&|\mathbf{n}(X)|_\mu^2 + \langle\mathbf{a}(X),\mathbf{b}(X)\rangle + \langle\mathbf{c}(X),\mathbf{l}(X)\rangle\\ &\quad= t_0 + t_1\cdot X + t_2\cdot X^2.\end{aligned} \tag{40}$$

We define the target polynomial as

$$f(X) = |\mathbf{n}(X)|_\mu^2 + \langle\mathbf{a}(X),\mathbf{b}(X)\rangle = t_0 + t_1\cdot X + t_2\cdot X^2. \tag{41}$$

Let $\mathbf{s} = (\mathbf{s}_G,\mathbf{s}_A,\mathbf{s}_B,\mathbf{s}_H)$, where $\mathbf{s}_G,\mathbf{s}_A,\mathbf{s}_B$ are sampled uniformly. Set

$$\mathbf{c}(X) = (0,0,0,0,\omega,X,0,0), \mathbf{l} = (l_0,l_1,l_2,l_3,0,r_{l,5},r_{l,6},r_{l,7})$$

for a random ω. To cancel the $t_1X + t_2X^2$ terms, write $\mathbf{s}_H = \mathbf{s}'_H + \mathbf{q}$, where

$$\mathbf{q} = (0,0,0,0,-\omega^{-1}t_1,-t_2,0,0).$$

We choose $\mathbf{s}'_H$ from the affine subspace satisfying

$$\langle\mathbf{c}(X),\mathbf{l} + \mathbf{s}'_H X\rangle = 0,$$

equivalently $\mathbf{s}'_{H,4} = -\omega^{-1}r_{l,5}$ and $\mathbf{s}'_{H,5} = 0$. Then

$$\langle\mathbf{c}(X),\mathbf{l} + \mathbf{s}_H X\rangle = -t_1X - t_2X^2.$$

Thus the commitments $T_1,T_2,\hat{t},\tau_x$ are unnecessary: after receiving ψ,κ,η,ω, the prover computes $\mathbf{s}_H = \mathbf{s}'_H + \mathbf{q}$ for the commitment S. The remaining protocol is the same as UniBit.

We provide the formal details in the full version of the paper.

Theorem 6. *UniBit$^+$ is an argument of knowledge for* $\mathbf{R}_{\text{RingCT}}$ *with statistical completeness, CWEE and sHVZK.*