Zero-Knowledge Protocol for Knowledge of Known Discrete Logarithms: Applications to Ring Confidential Transactions and Anonymous Zether

Li Lin
Digital Technologies, Ant Group
felix.ll@antgroup.com

Tian Qiu
The University of Sydney
tqiu4893@uni.sydney.edu.au

Xin Wang
Digital Technologies, Ant Group
wx352699@antgroup.com

Hailong Wang
Digital Technologies, Ant Group
whl383799@antgroup.com

Changzheng Wei
Digital Technologies, Ant Group
changzheng.wcz@antgroup.com

Ying Yan
Digital Technologies, Ant Group
fuying.yy@antgroup.com

Wei Wang
Digital Technologies, Ant Group
wei.wangwwei@antgroup.com

Wenbiao Zhao
Digital Technologies, Ant Group
wenbiao.zwb@antgroup.com

***Abstract*—The securities of a large fraction of zero-knowledge arguments of knowledge schemes rely on the discrete logarithm (DL) assumption or the discrete logarithm relation assumption, such as Bulletproofs (S&P 18) and compressed Σ-protocol (CRYPTO 20). At the heart of these protocols is an interactive proof of knowledge between a prover and a verifier showing that a Pedersen vector commitment $P = h^{\rho} \cdot \boldsymbol{g}^{\boldsymbol{x}}$ to a vector $\boldsymbol{x}$ satisfies multi-variate equations, where the DL relations among the vector of generators $\boldsymbol{g}$ are unknown. However, in some circumstances, the prover may know the DL relations among the generators, and the DL relation assumption no longer holds, such as ring signatures, ring confidential transactions (RingCT) and *K*-out-of-*N* proofs, which will make the soundness proof of these protocols infeasible.**

This paper is concerned with a problem called knowledge of known discrete logarithms (KKDL) that appears but has not been clearly delineated in the literature. Namely, it asks to prove a set of multi-exponent equalities, starting with the fact that the prover may know the DL relations among the generators of these equalities. Our contributions are threefold: (1) We propose a special honest-verifier zero-knowledge protocol for the problem. Using the Fiat-Shamir heuristic and the improved inner-product argument of Bulletproofs, the proof size of our protocol is logarithmic to the dimension of the vector.

(2) As applications, our protocol can be utilized to construct logarithmic-size RingCT securely which fixes the issues of Omniring (CCS 19), ring signatures (with signature size $2 \cdot \lceil \log_2(N) \rceil + 10$ for ring size N) and K-out-of-N proof of knowledge (with proof size $2 \cdot \lceil \log_2(N) \rceil + 14$) which achieves the most succinct proof size improving on previous results. Meanwhile, we propose the first account-based multi-receiver privacy scheme considering the sender's privacy with logarithmic proof size (to the best of our knowledge). (3) We describe an attack on RingCT-3.0 (FC 20) where an attacker can spend a coin of an arbitrary amount that never existed on the blockchain.

***Index Terms*—Zero-Knowledge Proof, Bulletproofs, Compressed Σ-Protocol, Knowledge of Known Discrete Logarithms, RingCT, K-out-of-N Proof of Knowledge, Omniring, Anonymous Zether, Cryptanalysis on RingCT-3.0**

1. Introduction

Zero-knowledge arguments of knowledge [1] have been widely used in many prominent applications, such as cryptocurrencies [2], [3], [4], [5], verifiable outsourced computation [6], [7], [8], [9], and anonymous credentials [10], [11], [12], [13]. A zero-knowledge proof of knowledge is a protocol that enables the prover to generate a proof for convincing the verifier of the validity of a particular statement without revealing anything else except the statement itself.

In [14], [15], Bünz et al. proposed a special honest-verifier zero-knowledge (SHVZK) proof without a trusted setup, called Bulletproofs, based on the techniques of Bootle et al. [16]. In [17], Attema and Cramer proposed a compressed Σ-protocol to strengthen Σ-protocol theory while shortening its communication complexity from linear to logarithmic. In [18], Attema et al. extended the compressed Σ-protocol of [17] from linear equations to general homomorphisms. At the heart of these protocols is an interactive proof of knowledge between a prover and a verifier showing that the Pedersen vector commitment [19] to a vector of length n satisfies multi-variate equations. More concretely, suppose $\mathbb{G}$ is a cyclic group of prime order p, and choose $\boldsymbol{g} = (g_1, \cdots, g_n) \in \mathbb{G}^n$ and $h \in \mathbb{G}$ with unknown discrete logarithm (DL) relations [20]. These protocols aim to convince the verifier that a vector $\boldsymbol{x} = (x_1, \cdots, x_n)$ hidden in the Pedersen vector commitment $P = h^{\rho} \cdot \boldsymbol{g}^{\boldsymbol{x}}$ satisfies some constraints (a multi-variate polynomial equation of

degree 2 for [15], linear equations for [17], and general homomorphisms for [18]) without leaking any information about $\boldsymbol{x}$ where $\rho \leftarrow_{\$} \mathbb{Z}_p$. After that, these protocols use the improved inner-product argument to reduce the sizes of the proofs from linear to logarithmic.

The securities of the protocols mentioned above rely either on the discrete logarithm assumption or the discrete logarithm relation assumption, i.e., the DL relations among the vector $\boldsymbol{g}$ should be unknown "in nature", e.g., using some cryptographic-secure hash-to-point functions [21] to deduce $\boldsymbol{g}$. The reason is that one should be able to deduce $\boldsymbol{x} = \boldsymbol{y}$ from $\boldsymbol{g}^{\boldsymbol{x}} = \boldsymbol{g}^{\boldsymbol{y}}$ in the soundness proof of the Bulletproofs-like protocols. However, the unknownness of the DL relations among $\boldsymbol{g}$ may not be guaranteed "in nature" when dealing with public keys or commitments generated from the same generators, e.g., the ring confidential transactions (RingCT) in the blockchain [3], [22], [23].

RingCT is the cryptographic core component of Monero [3], which is one of the largest privacy-preserving cryptocurrencies. In this private transaction, the sender takes a set of public keys as the input which were outputs of previous transactions. He proves that he knows the secret keys of some of the public keys without showing which ones and just claims they have not been used before. It is similar in spirit to linkable ring signatures and achieves anonymity and double-spending prevention simultaneously. But in the original RingCT construction [3], the transaction size is linear with the size of the ring.

1.1. Related Works and Issues

A natural idea for applying the techniques of Bulletproofs to RingCT is to replace $\boldsymbol{g}$ with the ring of public keys. However, it raises a new issue. In this setting, the public keys were generated and chosen arbitrarily by users and the unknownness of the DL relations among these public keys is not guaranteed by the cryptographic-secure assumptions. The user may know the DL relations among these public keys since they may know more than one private key. This will make the soundness proof of Bulletproofs-like protocol infeasible. Therefore, this method should be used very carefully. We will give an overall introduction to Bulletproofs and explain previous works that some of them are flawed.

1.1.1. Bulletproofs. From a higher perspective of Bulletproofs, if we want to prove that $\boldsymbol{b}_L = \boldsymbol{0}^n$, these n constraints are equivalent to one inner-product constraint $\langle \boldsymbol{b}_L, \boldsymbol{y}^n \rangle = 0$ where $y \in \mathbb{Z}_p$ is a random challenge chosen by the verifier. The probability that one can deduce $\boldsymbol{b}_L \neq \boldsymbol{0}^n$ from $\langle \boldsymbol{b}_L, \boldsymbol{y}^n \rangle = 0$ is n/p, which is negligible. Meanwhile, suppose we have m inner-product constraints by taking a random linear combination of these inner-product constraints using another challenge $z \in \mathbb{Z}_p$ from the verifier. In that case, one can convert these constraints into a single inner-product constraint. Hence, all the constraints can be proven at once. Based on these ideas and the improved inner-product argument, Bulletproofs are well suited for constructing an (aggregated) range proof protocol with logarithmic size in the witness size and an arithmetic circuit proof protocol with logarithmic size in the circuit size.

1.1.2. RingCT-3.0. In FC 2020, Yuen et al. proposed RingCT-3.0 [23] for blockchain confidential transactions to replace the RingCT-1.0 of Monero [24] with a shorter size and stronger security. It is based on Bulletproofs and enjoys the logarithmic proof size. Suppose there are N unspent coins, and $\boldsymbol{vk} = (vk_1, \cdots, vk_N)$ is the vector of corresponding verification keys (the real protocol also includes commitments of the amounts, we show the detailed one in Appendix B). Let $\hat{\boldsymbol{g}} = (\hat{g}_1, \cdots, \hat{g}_N)$ be a vector of generators with unknown DL relations, and $\mathtt{ind}$ be the index of the coin that belongs to the prover. The prover commits $vk_{\mathtt{ind}}, \hat{g}_{\mathtt{ind}}$ using $B_1 = h^{\alpha_1} \cdot vk_{\mathtt{ind}} \cdot \hat{g}_{\mathtt{ind}}^d$ and commits $\hat{g}_{\mathtt{ind}}$ using $B_2 = h^{\alpha_2} \cdot \hat{g}_{\mathtt{ind}}$, where $\alpha_1, \alpha_2 \in \mathbb{Z}_p$ are random numbers, and d is a challenge sent by the verifier at the beginning of the protocol. At the verification procedure, the prover uses B_2 and d to cancel $\hat{g}_{\mathtt{ind}}$ from B_1 and then convinces the verifier that he knows the private key $sk_{\mathtt{ind}}$ corresponding to $vk_{\mathtt{ind}}$.

Unfortunately, this scheme is vulnerable such that a malicious user can spend a coin that never existed. The attacker generates a random verification key $vk' = g^{sk'}$, then appends vk' and $vk_{\mathtt{ind}}$ to B_2 as $B_2 = h^{\alpha_1} \cdot \hat{g}_{\mathtt{ind}} \cdot (vk_{\mathtt{ind}} \cdot vk'^{-1})^{\frac{1}{d}}$. During the verification procedure, the attacker uses B_2 and d to cancel $vk_{\mathtt{ind}}$ from B_1 as well as $g_{\mathtt{ind}}$ and appends vk' to B_1. The attacker only needs to show that he knows sk' instead of $sk_{\mathtt{ind}}$, so he can "forge" a proof and pass the verification procedure without knowing any private key in $\boldsymbol{vk}$.

Although RingCT-3.0 shows a soundness proof in Appendix A of [25], it has some mistakes. The main drawback is that the components of B_2 are not deduced from a strict security proof (i.e., rewinding technique). The authors omit the discussion for B_2 and think that B_2 "defaults" to consist of $\hat{g}_{\mathtt{ind}}$ and h. Therefore, the attacker can pad vk' and $vk_{\mathtt{ind}}$ to B_2, and the verifier cannot notice this change due to the hiding property of the commitment scheme. We have noticed the author of [25] about this attack.

1.1.3. Omniring. In [22], Lai et al. proposed a fully-fledged RingCT scheme in the discrete logarithm setting, which also has a proof size logarithmic in the size of the ring inherited from Bulletproofs. Their zero-knowledge proofs are based on the Bulletproofs framework. The main difference between Omniring and Bulletproofs is the way to embed the set of verification keys $\boldsymbol{vk}$ into the commitment P and the way to extract the multi-exponent equalities about the private keys in the soundness proof. Briefly speaking, Omniring notices the public key issue, so it embeds $\boldsymbol{vk}$ into the generator vector $\boldsymbol{g}$ to keep the DL relations unknown, which avoids the problem regarding soundness. We take one relation $\boldsymbol{q}_{1,1}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_1} = 1$ as an example, where the DL relations among $\boldsymbol{q}_{1,1} \in \mathbb{G}^{n_1}$ and $\boldsymbol{q}_2 \in \mathbb{G}^{n_2}$ may be known to the prover. When $\boldsymbol{b}_L \in \mathbb{Z}_P^{n_1}$ is a binary vector with one component that is "1", $\boldsymbol{q}_{1,1} = \boldsymbol{vk}$, $\boldsymbol{q}_2 = \{g\}$ and

$\boldsymbol{\psi}_1 = \{-sk\}$, then $\boldsymbol{q}_{1,1}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_1} = 1$ is the relation of ring signature. For this relation, one can deduce the DL relation assumption of $\boldsymbol{g}_e = \boldsymbol{g} \circ (\boldsymbol{q}_{1,1} \| \boldsymbol{q}_2)^e$ from the DL relation assumption of $\boldsymbol{g}$, where $e \in \mathbb{Z}_p$ is a random challenge. Since $\boldsymbol{q}_{1,1}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_1} = 1$ is an equality, one can deduce that $P = h^\rho \cdot \boldsymbol{g}^{(\boldsymbol{b}_L \| \boldsymbol{\psi}_1)} = h^\rho \cdot \boldsymbol{g}_e^{(\boldsymbol{b}_L \| \boldsymbol{\psi}_1)}$. Therefore, the prover can commit $\boldsymbol{b}_L \| \boldsymbol{\psi}_1$ using $P = h^\rho \cdot \boldsymbol{g}^{(\boldsymbol{b}_L \| \boldsymbol{\psi}_1)}$ at the beginning of the protocol and then use $P = h^\rho \cdot \boldsymbol{g}_e^{(\boldsymbol{b}_L \| \boldsymbol{\psi}_1)}$ at the verification procedure after receiving e from the verifier. Using two different values of e, an extractor can be constructed to extract the above equality in the soundness proof of the protocol.

As a RingCT protocol, Omniring considers more relations rather than one ring-signature relation. Considering relations $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ for $1 \le i \le m$, we introduce two random challenges $e, v \in \mathbb{Z}_p$ from the verifier following the ideas of [22]. The challenge v gathers all equalities $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ into one, and the witnesses become $\boldsymbol{b}_L$ and $\boldsymbol{a} = \sum_{i=1}^{m} v^i \cdot \boldsymbol{\psi}_i$. The challenge e combines the gathered equality with the commitment P. One can construct an extractor that extracts the gathered equality using two different values of e and then extracts each equality using m different values of v in the soundness proof of the protocol.

However, when constructing the RingCT, the soundness proof of [22] is problematic. Taking the above version as an example. Since the randomness v is chosen at the beginning of the protocol before committing $\boldsymbol{b}_L$ in Section 5.1 of [22], one cannot use the normal rewinding technique to extract the witness $\boldsymbol{\psi}_i$ because the prover may choose $\boldsymbol{b}_L$ depending on v. Therefore, the authors use the Schwartz-Zippel lemma [26] to overcome this in Appendix D.2 of [22]. For the Schwartz-Zippel lemma, the polynomial coefficients should be selected before choosing the random variable. However, in their scheme, the random variable v is chosen before committing the coefficients, so one may choose the coefficients according to v. Hence, the way how [22] uses the Schwartz-Zippel lemma is problematic which makes the soundness proof infeasible.

1.1.4. K-out-of-N proof of knowledge. K-out-of-N proof of knowledge is a generalization of one-out-of-N proof of knowledge, i.e., a prover can convince the verifier that he knows the witnesses for some K-subset of N public statements without revealing the exact K-subset. One can apply the K-out-of-N proof of knowledge to threshold ring signatures [27], allowing only a significant enough subset to compute a valid signature. In [28], Diamond proposed a generalization of one-out-of-N proof of knowledge, called many-out-of-many proofs, which reduces the communication complexity of the Zether payment system [29]. However, this generalization considers a prover that claims to know the private keys of all public keys in one of the orbits of a public permutation of N public keys. Therefore, the protocol only works for permutations with orbits of equal size. Since the permutation is public and of this specific form, this protocol does not constitute a general K-out-of-N proof of knowledge. In [18], Attema et al. proposed an SHVZK protocol for the K-out-of-N proof of knowledge with logarithmic communication for general K and N using the compressed Σ-protocol. The proof size is $4 \cdot \lceil \log_2(2N - K + 1) \rceil - 1$, and the authors argue that one can reduce the size to $2 \cdot \lceil \log_2(2N - K + 1) \rceil + 3$ but it is restricted on pairing-friendly elliptic curves.

1.2. Our Contributions

In this paper, we abstract a general problem called knowledge of known discrete logarithms (KKDL) and propose a special honest-verifier zero-knowledge (SHVZK) protocol $\Pi_{\texttt{KKDL}}$. It can be used as a black box for proving the knowledge of many secret vectors even when the prover knows the DL relations among the given generators and still enjoys the logarithmic proof size.

This protocol finds applications in many topics, for example, RingCT [3], [22], [23], K-out-of-N proof of knowledge (threshold ring signatures) [18], [27], [28] and ring signature [30], [31]. These schemes share a common characteristic, which is the requirement to prove knowledge among a set of group elements generated from the same generators, e.g., public keys or commitments. This paper considers a general case of these schemes and proposes a zero-knowledge protocol $\Pi_{\texttt{KKDL}}$ to solve this problem. As a result, based on our $\Pi_{\texttt{KKDL}}$ protocol, we give secure and more efficient constructions of RingCT, K-out-of-N proof of knowledge, and ring signature.

Our $\Pi_{\texttt{KKDL}}$ protocol consists of `inner` and `outer` protocols with the *linear witnesses substitution for commitment* lemma. It is different from Bulletproofs and Omniring. We demonstrate the novelties of our work by showing the gaps between them.

- Admittedly, the basic security proof procedure of the `inner` protocol and the way to compress the proof size are borrowed from Bulletproofs. Nevertheless, our primary emphasis lies in addressing the case of known DL relations. It is a common issue in many scenarios, but is not covered by Bulletproofs.
- Omniring noticed the known DL issue and tried to address it. But their technique falls short of encompassing the entire RingCT protocol, and their construction poses problems as we discussed in Section 1.1.3. Different from it, we not only embed the ring public keys into the generator vector, but also design the `outer` protocol to fix the issues identified in Omniring. This involves dividing the input commitment *P* into two parts and committing the witness before selecting the randomness. Consequently, our $\Pi_{\texttt{KKDL}}$ protocol covers a broader range of relations and can be seamlessly applied (as a black box) in constructing RingCT.

1.2.1. KKDL Proofs. Let us describe the $\Pi_{\texttt{KKDL}}$ protocol more concretely. It is a protocol for proving that $m + 1$ secret vectors $\boldsymbol{b}_L \in \mathbb{Z}_p^{n_1}$ and $\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_m \in \mathbb{Z}_p^{n_2}$ satisfy m multi-exponent equalities $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ for $1 \le i \le m$. Note that the prover may know the DL relations among

$\boldsymbol{q}_{1,1}, \cdots, \boldsymbol{q}_{1,m} \in \mathbb{G}^{n_1}, \boldsymbol{q}_2 \in \mathbb{G}^{n_2}$, and $\boldsymbol{b}_L$ satisfies n_1 public quadratic relations and k public linear relations (the quadratic relations constraint each component of $\boldsymbol{b}_L$ to a specific value, e.g., "0" or "1", and the linear relations constraint the combinations of the components of $\boldsymbol{b}_L$, e.g., only one component of $\boldsymbol{b}_L$ is "1").

$\Pi_{\texttt{KKDL}}$ consists of two parts, i.e., the `inner` protocol and the `outer` protocol. We give a general protocol that applies to the Pedersen vector commitments and proves that a committed vector satisfies n_1 public quadratic relations and k public linear relations for the `inner` protocol. The `inner` protocol is an extension of Bulletproofs. There is a gap between the input commitment $P = h^\rho \cdot \boldsymbol{g}^{\boldsymbol{b_L}}$ and the commitment $P = h^\rho \cdot \boldsymbol{g}^{\boldsymbol{b_L}} \cdot \boldsymbol{h}^{\boldsymbol{b_R}}$ of the Bulletproofs-like proof system for the public quadratic relations. We propose the *linear witnesses substitution for commitment* lemma to fill the gap. Since the `inner` protocol only supports the generators $\boldsymbol{g}$ of unknown DL relations and cannot support multi-exponent equalities $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ with generators of known DL relations, we propose the `outer` protocol for embedding these equalities into $\boldsymbol{g}$ just as how $\boldsymbol{g}$ becomes $\boldsymbol{g}_e$ in the former part of this section. Using the Fiat-Shamir heuristic [32], we can convert $\Pi_{\texttt{KKDL}}$ into a non-interactive protocol that is secure and full zero-knowledge in the random oracle model. Finally, we compress the proof size of $\Pi_{\texttt{KKDL}}$ from linear to logarithmic using the improved inner-product argument from [15], [17]. The proof size is $2 \cdot \lceil \log_2(2n_1 + n_2) \rceil + 2 \cdot \lceil \log_2(n_2 + 1) \rceil + 9$.

1.2.2. RingCT. As we mentioned in Section 1.1.3, there are some issues in previous Bulletproofs-based RingCT. By implementing our protocol in RingCT, we get a secure RingCT scheme without encountering any previous issues, while still enjoying logarithmic communication cost. To overcome those problems, note that P is the commitment of $(\boldsymbol{a}, \boldsymbol{b}_L)$ and $\boldsymbol{a}$ is based on v, we split the commitment into two parts. Firstly, the prover commits $\boldsymbol{b}_L$ at the beginning of the protocol before receiving v. Secondly, the prover commits $\boldsymbol{a}$ after receiving v and adds an in-line Σ-protocol to ensure that this commitment does not contain any information about $\boldsymbol{b}_L$. Then, P is the combination of these two commitments. These methods solve the soundness problem of Omniring since $\boldsymbol{b}_L$ is fixed at the beginning of the protocol and one can rewind v many times in the soundness proof because v is chosen after committing $\boldsymbol{b}_L$. Section 3.2 and 4 gives a more detailed description. The proof size of the scheme is $2 \cdot \lceil \log_2(3 + |\mathcal{R}| + |\mathcal{R}||\mathcal{S}| + \beta|\mathcal{T}| + 3|\mathcal{S}|) \rceil + 2\lceil \log_2(4 + |\mathcal{R}|) \rceil +$ 12 elements [1], where $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$ are the size of the ring, the set of source accounts, and the set of target accounts in a transaction respectively, and 2^β is the maximum currency amount that can be sent in a single transaction. Since the proof size is logarithm, our scheme enjoys a shorter size than all the previous RingCT schemes *without* trusted setup and pairing-friendly elliptic curves to the best of our knowledge. We compare our scheme with prior works without a trusted setup in Table 1.

1.2.3. Anonymous Zether. In [28], Diamond proposed an account-based privacy-preserving protocol for blockchain transactions called Anonymous Zether. The way for anonymous is that a sender may hide herself and the receivers in a larger ring $\vec{\boldsymbol{R}} = \{pk_i, 0 \leq i \leq N-1\}$. The original paper [28] only consider the case that the number of receiver is one, but it may be larger than one. In this paper, we consider the multi-receiver scenario. Although [34] also considers the multi-receiver scenario, they do not consider the anonymity of sender since an artifact of Ethereum where invocation to smart contract trivially reveals the identity of the invoking party. However, the identity of the invoking party can be different from the identity of the smart contract, and we can trivially use Tor network [35], paymaster [36] and ring signature to avoid this. The proof size is $2 \cdot \lceil \log_2(5 + 10|\mathcal{R}|) \rceil + 2\lceil \log_2(5 + 7|\mathcal{R}| + (|\mathcal{R}| + 1)\beta) \rceil + 9$ elements. The term $(|\mathcal{R}|+1)\beta$ arises because there are more than one receivers. For the case of one receiver, we have better proof size compared to [28].

1.2.4. K-out-of-N proof of knowledge. We construct an SHVZK protocol for the K-out-of-N proof of knowledge using $\Pi_{\texttt{KKDL}}$, and the proof size of our scheme is $2 \cdot \lceil \log_2(N) \rceil + 14$ elements. To the best of our knowledge, our scheme enjoys a shorter size than the previous scheme *without* pairing-friendly elliptic curves, and we compare our scheme with prior works in Table 1. The reason why our scheme enjoys a shorter size is that we use the improved inner-product argument to compress the vector of dimension $2N$ to $2 \cdot \log_2(N)$, while the compression mechanism of [18] can only compress it to $4 \cdot \log_2(N)$.

It also implies a ring signature scheme as follows. Firstly, we build an SHVZK protocol for the one-out-of-N proof of knowledge [38] using $\Pi_{\texttt{KKDL}}$. This protocol can be converted to a signature of knowledge [39] which implies a ring signature scheme. The signature size of our scheme is $2 \cdot \lceil \log_2(N) \rceil + 10$ elements, where N is the number of verification keys in the ring. The signature size is almost the same as the state-of-the-art construction from [22].

[1] One point from $\mathbb{G}$ can be stored as 32 bytes plus one bit in the compressed form when $\mathbb{G}$ is the elliptic curve secp256r1 [33], and one element from $\mathbb{Z}_p$ can be stored as 32 bytes, so we measure them as the same.

[2] [37] also has logarithmic proof size, but it is less efficient than ours with the same anonymous set since it requires separate rings for separate source accounts. Meanwhile, the separated range proof system will incur unnecessary computational and communication overheads.

Type	Scheme	Size
RingCT	RingCT-1.0 [24]	$(\lvert\mathcal{R}\rvert+2)(\lvert\mathcal{S}\rvert+1)+\lceil\log_2(\beta\lvert\mathcal{T}\rvert)\rceil$
RingCT	PBT [37] [2]	$(\lvert\mathcal{S}\rvert+1)(7\log_2(\lvert\mathcal{R}\rvert+3)+\lceil\log_2(\beta\lvert\mathcal{T}\rvert)\rceil$
RingCT	Section 4	$2\lceil\log_2(3+\lvert\mathcal{R}\rvert+\lvert\mathcal{R}\rvert\lvert\mathcal{S}\rvert+\beta\lvert\mathcal{T}\rvert+3\lvert\mathcal{S}\rvert)\rceil+2\lceil\log_2(4+\lvert\mathcal{R}\rvert)\rceil+12$
K-out-of-N	[18]	$4\cdot\lceil\log_2(2N-K+1)\rceil-1$
K-out-of-N	Section 5	$2\cdot\lceil\log_2(N)\rceil+14$

TABLE 1: Efficiency comparisons between our instantiations and the most efficient RingCT/ K-out-of-N proof of knowledge schemes. [3]

1.2.5. Cryptoanalysis of RingCT-3.0. As another contribution of this paper, we give an attack on RingCT-3.0 where a malicious user can spend a coin of an arbitrary amount that never existed on the chain. We have discussed it briefly in Section 1.1.2 and will show the detailed cryptanalysis in Appendix B. Note that in our protocol, we do not use such cancellation techniques of RingCT-3.0. Meanwhile, we add an in-line Σ-protocol in the `outer` protocol to ensure that the commitment does not contain any other generator.

1.3. Organization

The rest of the paper is organized as follows. We provide the definitions of commitments and zero-knowledge arguments of knowledge, along with notations in Section 2. Section 3 gives the definition, the construction, and the security proof of $\Pi_{\texttt{KKDL}}$. Section 4 proposes a protocol for RingCT using $\Pi_{\texttt{KKDL}}$. Section 5 proposes a protocol for the K-out-of-N proof of knowledge using $\Pi_{\texttt{KKDL}}$. Meanwhile, Appendix A provides the detailed construction of multi-receiver Anonymous Zether. Appendix B provides the detailed cryptanalysis of RingCT-3.0. Appendix D constructs a ring signature scheme using $\Pi_{\texttt{KKDL}}$.

2. Preliminaries

This section gives the notations used in this paper and reviews some underlying tools.

2.1. Notations

Let λ denote a security parameter, and p be a prime number of length λ. Let $\mathbb{G}$ denote a cyclic group of prime order p, $\mathbb{Z}_p$ denote the ring of integers modulo p, and $\mathbb{Z}_p^*$ denote $\mathbb{Z}_p \setminus \{0\}$. Let $(\mathbb{G}, p) \leftarrow \textbf{GroupGen}(1^\lambda)$ be an algorithm that inputs a security parameter λ and outputs a cyclic group $\mathbb{G}$ of prime order p. $x \leftarrow_\$ \mathbb{Z}_p$ denotes the uniform sampling of an element from $\mathbb{Z}_p$. Throughout this paper, we will use bold letters to denote vectors, i.e., $\boldsymbol{a} \in \mathbb{Z}^n$ is a vector with elements $a_1, a_2, \cdots, a_n \in \mathbb{Z}$. Let $\boldsymbol{a}, \boldsymbol{b}$ denote vectors from $\mathbb{Z}_p$, and $\boldsymbol{G}, \boldsymbol{H}$ denote vectors from $\mathbb{G}$.

For two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_p^n$, the inner product between $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined as $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum_{i=1}^n a_i \cdot b_i \in \mathbb{Z}_p$, and the Hadamard product between $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined as $\boldsymbol{a} \circ \boldsymbol{b} = (a_1 \cdot b_1, \cdots, a_n \cdot b_n) \in \mathbb{Z}_p^n$. For a scalar $c \in \mathbb{Z}_p$ and a vector $\boldsymbol{a} \in \mathbb{Z}_p^n$, the scalar multiplication is denoted as $c \cdot \boldsymbol{a} = (c \cdot a_1, \cdots, c \cdot a_n) \in \mathbb{Z}_p^n$. For $\boldsymbol{a} \in \mathbb{Z}_p^n$ and $\boldsymbol{G} \in \mathbb{G}^n$, the multi-exponent is denoted as $\boldsymbol{G}^{\boldsymbol{a}} = \prod_{i=1}^n G_i^{a_i}$, and $\boldsymbol{G}^{\circ \boldsymbol{a}} = (G_1^{a_1}, \cdots, G_n^{a_n})$. Meanwhile, for $k \in \mathbb{Z}_p^*$, we use $\boldsymbol{k}^n$ to denote the vector containing the first n powers of k, i.e., $\boldsymbol{k}^n = (1, k, k^2, \cdots, k^{n-1})$. For example, $\boldsymbol{2}^n = (1, 2, 4, \cdots, 2^{n-1})$ and $\boldsymbol{k}^{-n} = (1, k^{-1}, \cdots, k^{-n+1})$. For $\boldsymbol{g} \in \mathbb{G}^n$ and $c \in \mathbb{Z}_p$, $\boldsymbol{g}^c = (g_1^c, \cdots, g_n^c)$. For two vectors $\boldsymbol{a} \in \mathbb{Z}_p^n$ and $\boldsymbol{b} \in \mathbb{Z}_p^m$, the concatenation of $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted as $\boldsymbol{a} \| \boldsymbol{b} \in \mathbb{Z}_p^{m+n}$.

2.2. Basic Assumptions

The security of our protocol depends on the discrete logarithm (DL) assumption and the discrete logarithm relation assumption defined in Definition 1 and 2, respectively.

Definition 1 (Discrete Logarithm Assumption)**.** *We say that the discrete logarithm assumption holds relative to* **GroupGen** *if for all non-uniform polynomial-time adversaries $\mathcal{A}$, there exists a negligible function $\mu(\lambda)$ such that*

$$\Pr\left[g^x = h \;\middle|\; \begin{array}{c}(\mathbb{G}, p) \leftarrow \textbf{GroupGen}(1^\lambda), \\ g, h \leftarrow_\$ \mathbb{G}, x \leftarrow \mathcal{A}(\mathcal{G}, p, g, h)\end{array}\right] < \mu(\lambda).$$

Definition 2 (Discrete Logarithm Relation Assumption)**.** *We say that the discrete logarithm relation assumption holds with respect to* **GroupGen** *if for all $n \geq 1$ and all non-uniform polynomial-time adversaries $\mathcal{A}$, there exists a negligible function $\mu(\lambda)$ such that*

$$\Pr\left[\begin{array}{c}(\exists a_i \neq 0 \text{ for} \\ i \in [1, n]) \wedge \\ (\prod_{i=1}^n g_i^{a_i} = 1)\end{array} \;\middle|\; \begin{array}{c}(\mathbb{G}, p) \leftarrow \textbf{GroupGen}(1^\lambda), \\ g_1, \cdots, g_n \leftarrow_\$ \mathbb{G} \\ \{a_i\}_{i=1}^n \leftarrow \mathcal{A}(\mathbb{G}, p, \{g_i\}_{i=1}^n)\end{array}\right] < \mu(\lambda).$$

2.3. Pedersen Vector Commitment

A commitment scheme consists of three algorithms: **Setup**, **Com**, and **Open**. The setup algorithm outputs the

[3] Although [40] gives a RingCT scheme, we do not consider it here due to the following two reasons: (1) In Section V.C of [40], it aggregates steps 19 and 20 to shorten the proof size. However, one cannot distinguish equations (19) and (20) from the aggregated verification equation in the soundness proof. Meanwhile, $\boldsymbol{g}$ should be sent at the beginning of the protocol as Omniring instead of fixing at the beginning to make the DL relation with the public keys unknown. (2) In Section VI.B of [40], they use u to gather public keys, tags and prime tags together to form the RingCT protocol. However, the way to generate u follows Omniring and RingCT-3.0 which may face the same problems as we analyzed in this paper.

public parameters pp for this scheme inputting the security parameter λ. The commit algorithm **Com** is a function $M_{pp} \times R_{pp} \to C_{pp}$, where M_{pp}, R_{pp}, and C_{pp} is the message space, the randomness space, and the commitment space, respectively. To commit a message $v \in M_{pp}$, the sender computes $C \leftarrow \mathbf{Com}_{pp}(v, r)$ by choosing $r \leftarrow_{\$} R_{pp}$. The open algorithm **Open** is a function $C_{pp} \times M_{pp} \times R_{pp} \to \{0, 1\}$. To de-commit a message $v \in M_{pp}$, the sender sends v and r to the receiver. The receiver outputs "1" if $C = \mathbf{Com}_{pp}(v, r)$ and "0" otherwise.

In this paper, we use Pedersen vector commitment [19], where $M_{pp} = \mathbb{Z}_p^n$, $R_{pp} = \mathbb{Z}_p$, and $C_{pp} = \mathbb{G}$. The setup algorithm works as follows: $(\mathbb{G}, p, \boldsymbol{g} = (g_1, \cdots, g_n), h) \leftarrow \mathbf{Setup}(1^\lambda)$, where $(\mathbb{G}, p) \leftarrow \mathbf{GroupGen}(1^\lambda)$ and $g_1, \cdots, g_n, h \leftarrow_{\$} \mathbb{G}$ with unknown DL relations. The commit algorithm works as follows: $C \leftarrow \mathbf{Com}_{pp}(\boldsymbol{v} = (v_1, \cdots, v_n), r)$, where $C = h^r \prod_{i=1}^{n} g_i^{v_i} \in \mathbb{G}$. The open algorithm outputs "1" if $C = h^r \prod_{i=1}^{n} g_i^{v_i}$ and "0" otherwise.

2.4. Zero-Knowledge Arguments of Knowledge

A zero-knowledge argument of knowledge is a two-party protocol between a prover and a verifier. The prover tries to convince the verifier that a statement holds without revealing any information about the witness. This proof system consists of three probabilistic polynomial-time algorithms **Setup**, $\mathcal{P}$, and $\mathcal{V}$. The setup algorithm outputs a common reference string σ on inputting a security parameter λ. The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ are interactive algorithms. The transcript produced by $\mathcal{P}$ and $\mathcal{V}$ when interacting on inputs x and y is denoted by $tr \leftarrow \langle \mathcal{P}, \mathcal{V} \rangle$. As the output of this protocol, we use the notation $\langle \mathcal{P}, \mathcal{V} \rangle = b$, where $b = 1$ if $\mathcal{V}$ accepts and $b = 0$ if $\mathcal{V}$ rejects. The proof is *public coin* if an honest verifier generates his responses to $\mathcal{P}$ uniformly.

Let $\mathcal{R}$ be a polynomial-time verifiable ternary relation for common reference string σ, statement x, and witness w, and let $\mathcal{L}$ be the corresponding language, i.e., $\mathcal{L} = \{x \mid \exists w, \text{ s.t., } (\sigma, x, w) \in \mathcal{R}\}$. The argument of knowledge is defined as follows.

Definition 3 (Argument of Knowledge)**.** *The triple* $(\mathbf{Setup}, \mathcal{P}, \mathcal{V})$ *is called an argument of knowledge for the relation* $\mathcal{R}$ *if it satisfies the following two definitions.*

Definition 4 (Perfect Completeness)**.** $(\mathbf{Setup}, \mathcal{P}, \mathcal{V})$ *has perfect completeness if for all non-uniform polynomial-time interactive adversaries* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|c} \langle \mathcal{P}(\sigma, x, w), \mathcal{V}(\sigma, x) \rangle = 1 \\ \vee\ (\sigma, x, w) \notin \mathcal{R} \end{array} \middle| \begin{array}{c} \sigma \leftarrow \mathbf{Setup}(1^\lambda) \\ (x, w) \leftarrow \mathcal{A}(\sigma) \end{array}\right] = 1.$$

Definition 5 (Computational Witness-Extended Emulation)**.** $(\mathbf{Setup}, \mathcal{P}, \mathcal{V})$ *has witness-extended emulation if for any deterministic polynomial-time prover* $\mathcal{P}^*$*, there exists an expected polynomial-time emulator* $\mathcal{E}$ *such that for all non-uniform polynomial-time interactive adversaries* $\mathcal{A}$*, there exists a negligible function* $\mu(\lambda)$ *such that the difference between the following two probabilities is smaller than* $\mu(\lambda)$*.*

$$\Pr\left[\mathcal{A}(tr) = 1 \;\middle|\; \begin{array}{c} \sigma \leftarrow \mathbf{Setup}(1^\lambda);\ (x, s) \leftarrow \mathcal{A}(\sigma); \\ tr \leftarrow \langle \mathcal{P}^*(\sigma, x, s), \mathcal{V}(\sigma, x) \rangle \end{array}\right] \text{ and}$$

$$\Pr\left[\begin{array}{c} \mathcal{A}(tr) = 1 \wedge \\ \text{if } tr \text{ is accepting,} \\ \text{then } (\sigma, x, w) \in \mathcal{R} \end{array} \middle| \begin{array}{c} \sigma \leftarrow \mathbf{Setup}(1^\lambda);\ (x, s) \leftarrow \mathcal{A}(\sigma); \\ (tr, w) \leftarrow \mathcal{E}^{\langle \mathcal{P}^*(\sigma, x, s), \mathcal{V}(\sigma, x) \rangle}(\sigma, x) \end{array}\right],$$

where $\mathcal{E}$ *has access to the oracle* $\langle \mathcal{P}^*(\sigma, x, s), \mathcal{V}(\sigma, x) \rangle$ *that permits rewinding to a specific round and rerunning with* $\mathcal{V}$ *using fresh randomness.*

We use the witness-extended emulation to define knowledge soundness as in [15], and the value s can be regarded as the state of $\mathcal{P}^*$ including the randomness. Whenever an adversary produces an argument that can pass the verification with some probability, an emulator can produce an identically distributed argument (i.e., witness) with the same probability. The way the emulator produces such an argument is to rewind the interaction between the prover and the verifier, the internal state of the prover is the same. Still, the randomness of the verifier is fresh.

The protocols in this paper require the zero-knowledge property. We use special honest-verifier zero-knowledge (SHVZK), i.e., given the verifier's challenge values, it is possible to simulate the entire argument without knowing the witness efficiently.

Definition 6 (Perfect Special Honest-Verifier Zero-Knowledge)**.** *A public coin argument* $(\mathbf{Setup}, \mathcal{P}, \mathcal{V})$ *is perfect special honest-verifier zero-knowledge (SHVZK) for* $\mathcal{R}$ *if there exists probabilistic polynomial-time simulator* S *such that for all non-uniform polynomial-time interactive adversaries* $\mathcal{A}$*,*

$$\begin{aligned} &\Pr\left[\begin{array}{c} \mathcal{A}(tr) = 1 \wedge \\ (\sigma, x, w) \in \mathcal{R} \end{array} \middle| \begin{array}{c} \sigma \leftarrow \mathbf{Setup}(1^\lambda); \\ (x, w, \rho) \leftarrow \mathcal{A}(\sigma); \\ tr \leftarrow \langle \mathcal{P}(\sigma, x, w), \mathcal{V}(\sigma, x, \rho) \rangle \end{array}\right] \\ =&\Pr\left[\begin{array}{c} \mathcal{A}(tr) = 1 \wedge \\ (\sigma, x, w) \in \mathcal{R} \end{array} \middle| \begin{array}{c} (x, w, \rho) \leftarrow \mathcal{A}(\sigma); \\ tr \leftarrow S(x, \rho) \end{array}\right] \end{aligned}$$

where ρ *is the public coin randomness used by* $\mathcal{V}$*.*

3. General Proof System for Knowledge of Known Discrete Logarithms

In this section, we construct an interactive protocol for proving that $m + 1$ secret vectors $\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_m \in \mathbb{Z}_p^{n_2}$ and $\boldsymbol{b}_L \in \mathbb{Z}_p^{n_1}$ satisfy m multi-exponent relations $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ for $1 \le i \le m$, where the DL relations among $\boldsymbol{q}_{1,1}, \cdots, \boldsymbol{q}_{1,m} \in \mathbb{G}^{n_1}, \boldsymbol{q}_2 \in \mathbb{G}^{n_2}$ may be known to the prover and $\boldsymbol{b}_L$ also satisfies n_1 public quadratic relations and k public linear relations. More concretely, we construct an SHVZK protocol for knowledge of known discrete logarithms (KKDL) problem, i.e., we give an SHVZK protocol for the following relation

$$\mathcal{R}_{\text{KKDL}} = \left\{ \begin{array}{c} (d_1, \cdots, d_k, \\ \phi_1, \cdots, \phi_k, \\ \boldsymbol{q}_{1,1}, \cdots, \\ \boldsymbol{q}_{1,m}, f, \boldsymbol{q}_2) \end{array} \middle| \begin{array}{c} \exists\ \boldsymbol{b}_L, \boldsymbol{\psi}_i, \text{ s.t., } \boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1 \\ \text{for } 1 \le i \le m \wedge f(\boldsymbol{b}_L) = \boldsymbol{0}^{n_1} \\ \wedge\ \phi_j(\boldsymbol{b}_L) = d_j \text{ for } 1 \le j \le k \end{array} \right\}, \tag{1}$$

where $d_j \in \mathbb{Z}_p$ for $1 \le j \le k$. The function $f : \mathbb{Z}_p^{n_1} \to \mathbb{Z}_p^{n_1}$ is a quadratic function such that $f(\boldsymbol{b}_L) = \boldsymbol{\alpha} \circ \boldsymbol{b}_L \circ \boldsymbol{b}_L + \boldsymbol{\beta} \circ \boldsymbol{b}_L + \boldsymbol{\gamma}$ where $\boldsymbol{\alpha} \in (\mathbb{Z}_p^*)^{n_1}$, $\boldsymbol{\beta} \in \mathbb{Z}_p^{n_1}$ and $\boldsymbol{\gamma} \in \mathbb{Z}_p^{n_1}$ are public coefficients, and $\phi_j : \mathbb{Z}_p^{n_1} \to \mathbb{Z}_p$ is a linear function such that $\phi_j(\boldsymbol{b}_L) = \langle \boldsymbol{b}_L, \boldsymbol{\zeta}_j \rangle$ where $\boldsymbol{\zeta}_j \in \mathbb{Z}_p^{n_1}$ is a public vector.

We split the protocol for $\mathcal{R}_{\texttt{KKDL}}$ into two parts, i.e., the `inner` protocol and the `outer` protocol. The `inner` protocol (Section 3.1) gives a protocol that applies to Pedersen vector commitments and proves that a committed vector satisfies n_1 public quadratic relations and k public linear relations. Since the `inner` protocol can only support the generators $\boldsymbol{g}$ of unknown DL relations and cannot handle multi-exponent equalities $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ with generators of known DL relations, we propose the `outer` protocol to embed these equalities into $\boldsymbol{g}$. Informally, one can deduce the DL relation assumption of $\boldsymbol{g}_e = \boldsymbol{g} \circ (\boldsymbol{q}_{1,1} || \boldsymbol{q}_2)^e$ from the DL relation assumption of $\boldsymbol{g}$, where $e \in \mathbb{Z}_p$ is a random challenge from the verifier. Since $\boldsymbol{q}_{1,1}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_1} = 1$ is an equality, one can deduce that $P = h^\rho \cdot \boldsymbol{g}^{(\boldsymbol{b}_L \| \boldsymbol{\psi}_1)} = h^\rho \cdot \boldsymbol{g}_e^{(\boldsymbol{b}_L \| \boldsymbol{\psi}_1)}$. Then, the `inner` protocol can be used to handle the commitment P with generators $\boldsymbol{g}_e$. After that, we provide the `outer` protocol (Section 3.2), which proves $\mathcal{R}_{\texttt{KKDL}}$ using the `inner` protocol as a sub-protocol. In Section 3.3, we analyze the securities of the `inner` protocol and the `outer` protocol. Section 3.4 compresses the proof size from linear to logarithmic using the improved inner-product argument from [15] and [17].

3.1. The Inner Protocol

In this section, we generalize the ideas of Bulletproofs and construct an interactive protocol for proving that two secret vectors $\boldsymbol{b}_L \in \mathbb{Z}_p^{n_1}$ and $\boldsymbol{a} \in \mathbb{Z}_p^{n_2}$ satisfy a public vector commitment $P = h^{\rho_L} \cdot \boldsymbol{g}_1^{\boldsymbol{b}_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \in \mathbb{G}$, a public quadratic function (n_1 quadratic relations) $f : \mathbb{Z}_p^{n_1} \to \mathbb{Z}_p^{n_1}$ and public linear functions $\phi_i : \mathbb{Z}_p^{n_1} \to \mathbb{Z}_p$ for $1 \le i \le k$ without leaking any information about $\boldsymbol{b}_L$ and $\boldsymbol{a}$. More concretely, we want to construct a zero-knowledge protocol for the following relation (we treat $d_1, \cdots, d_k$ as public values while Bulletproofs treat them as private witnesses):

$$\mathcal{R}_{\texttt{inner}} = \left\{ \begin{array}{c|c} (\boldsymbol{g}_1, \boldsymbol{g}_2, h, P, & \exists\ \boldsymbol{b}_L, \boldsymbol{a}, \rho_L, \text{ s.t., } f(\boldsymbol{b}_L) = \boldsymbol{0}^{n_1} \\ f, d_1, \cdots, d_k, & \wedge P = h^{\rho_L} \boldsymbol{g}_1^{\boldsymbol{b}_L} \boldsymbol{g}_2^{\boldsymbol{a}} \\ \phi_1, \cdots, \phi_k) & \wedge \phi_i(\boldsymbol{b}_L) = d_i \text{for } 1 \le i \le k \end{array} \right\} \tag{2}$$

where $P \in \mathbb{G}$, $\boldsymbol{g}_1 \in \mathbb{G}^{n_1}$, $\boldsymbol{g}_2 \in \mathbb{G}^{n_2}$, $\rho_L \in \mathbb{Z}_p$, $\boldsymbol{b}_L \in \mathbb{Z}_p^{n_1}$, $\boldsymbol{a} \in \mathbb{Z}_p^{n_2}$ and $d_i \in \mathbb{Z}_p$ for $1 \le i \le k$. The DL relations among $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, g and h are unknown.

Attema et al. give Σ-protocols for the following relation:

$$\mathcal{R}_f = \left\{ \ (P \in \mathbb{G}, y) \ \middle| \ \begin{array}{c} \exists\ \boldsymbol{b}_L \in \mathbb{Z}_p^{n_1}, \rho \in \mathbb{Z}_p, \text{ s.t.,} \\ P = h^\rho \cdot \boldsymbol{g}_1^{\boldsymbol{b}_L} \wedge y = \phi(\boldsymbol{b}_L) \end{array} \right\}, \tag{3}$$

where $y \in \mathbb{Z}_p$, $\phi : \mathbb{Z}_p^{n_1} \to \mathbb{Z}_p$ is a linear function (Section 3.1 of [17]) and $y \in \mathbb{G}$, $\phi : \mathbb{Z}_p^{n_1} \to \mathbb{G}$ is a group homomorphic function (Section 3.1 of [18]). The differences between (2) and (3) are that (2) is more expressive since it supports the quadratic function and more than one linear functions. Meanwhile, (2) is only an inner relation, and we will make it more expressive by adding equalities of known DL relations related to $\boldsymbol{b}_L$ in Section 3.2.

Rewrite $f(\boldsymbol{b}_L) = \boldsymbol{b}_L \circ (\boldsymbol{\alpha} \circ \boldsymbol{b}_L + \boldsymbol{\beta}) + \boldsymbol{\gamma}$. Construct another vector $\boldsymbol{b}_R$ that satisfies

$$\boldsymbol{b}_R = \boldsymbol{\alpha} \circ \boldsymbol{b}_L + \boldsymbol{\beta} \in \mathbb{Z}_p^{n_1}. \tag{4}$$

Then, one can rewrite $f(\boldsymbol{b}_L) = \boldsymbol{0}^{n_1}$ as

$$\boldsymbol{b}_L \circ \boldsymbol{b}_R = -\boldsymbol{\gamma}. \tag{5}$$

Since $\boldsymbol{b}_R = \boldsymbol{\alpha} \circ \boldsymbol{b}_L + \boldsymbol{\beta}$ and $P = h^{\rho_L} \cdot \boldsymbol{g}_1^{\boldsymbol{b}_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}}$, one can transform the commitment of $\boldsymbol{b}_L$ into the commitment of $\boldsymbol{b}_L$ and $\boldsymbol{b}_R$ using the following lemma.

Lemma 1 (Linear Witnesses Substitution for Commitment)**.** *Given one vector of generators $\boldsymbol{h} \in \mathbb{G}^{n_1}$ where the DL relations among $\boldsymbol{h}$, $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, g, and h are unknown, $\tilde{P} = P \cdot \boldsymbol{h}^{\boldsymbol{\beta}}$ is a commitment of $\boldsymbol{b}_L$ and $\boldsymbol{b}_R$ using $\boldsymbol{g} = \boldsymbol{g_1} \circ \boldsymbol{h}^{\circ -\boldsymbol{\alpha}} \in \mathbb{G}^{n_1}$ and $\boldsymbol{h}$ as generators.*

Proof. By the definitions of $\boldsymbol{b}_R$ and P, one has that

$$\begin{aligned} \tilde{P} &= P \cdot \boldsymbol{h}^{\boldsymbol{\beta}} = h^{\rho_L} \cdot \boldsymbol{g}_1^{\boldsymbol{b}_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \cdot \boldsymbol{h}^{\boldsymbol{\beta}} \\ &= h^{\rho_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \cdot (\boldsymbol{g}_1 \circ \boldsymbol{h}^{\circ -\boldsymbol{\alpha}})^{\boldsymbol{b}_L} \cdot \boldsymbol{h}^{\boldsymbol{\alpha} \circ \boldsymbol{b}_L + \boldsymbol{\beta}} \\ &= h^{\rho_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \cdot \boldsymbol{g}^{\boldsymbol{b}_L} \cdot \boldsymbol{h}^{\boldsymbol{b}_R}. \end{aligned} \tag{6}$$

Since the DL relations among $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{h}$, g, and h are unknown and $\boldsymbol{\alpha} \in (\mathbb{Z}_p^*)^{n_1}$, the DL relations among $\boldsymbol{g} = \boldsymbol{g_1} \circ \boldsymbol{h}^{\circ -\boldsymbol{\alpha}}$, $\boldsymbol{g}_2$, $\boldsymbol{h}$, g, and h are unknown. Therefore, $\tilde{P}$ is a commitment of $\boldsymbol{b}_L$ and $\boldsymbol{b}_R$. □

By Lemma 1, $\mathcal{R}_{\texttt{inner}}$ is equivalent to $\tilde{\mathcal{R}}_{\texttt{inner}}$ defined as follows:

$$\tilde{\mathcal{R}}_{\texttt{inner}} = \left\{ \begin{array}{c|c} (\boldsymbol{g}, \boldsymbol{g}_2, \boldsymbol{h}, h, & \exists\ \boldsymbol{b}_L, \boldsymbol{b}_R, \boldsymbol{a}, \rho_L, \text{ s.t.,} \\ d_1, \cdots, d_k, & \tilde{P} = h^{\rho_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \cdot \boldsymbol{g}^{\boldsymbol{b}_L} \cdot \boldsymbol{h}^{\boldsymbol{b}_R} \\ \tilde{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, & \wedge\ \boldsymbol{b}_L \circ \boldsymbol{b}_R = -\boldsymbol{\gamma} \\ \phi_1, \cdots, \phi_k) & \wedge\ \boldsymbol{b}_R = \boldsymbol{\alpha} \circ \boldsymbol{b}_L + \boldsymbol{\beta} \\ & \wedge\ \phi_i(\boldsymbol{b}_L) = d_i \text{ for } 1 \le i \le k \end{array} \right\} \tag{7}$$

A proof system for $\tilde{\mathcal{R}}_{\texttt{inner}}$ gives a proof system for $\mathcal{R}_{\texttt{inner}}$. Hence, it suffices to provide a proof system for $\tilde{\mathcal{R}}_{\texttt{inner}}$. In the following part of this section, we will give an SHVZK protocol $\tilde{\Pi}_{\texttt{inner}}$ for $\tilde{\mathcal{R}}_{\texttt{inner}}$, which is also an SHVZK protocol for $\mathcal{R}_{\texttt{inner}}$.

Next, we will convert $2n_1$ constraints of (4) and (5) as a single inner-product constraint. Using randomness $y \in \mathbb{Z}_p^*$ from the verifier and the Schwartz–Zippel lemma [41], the prover can prove that (4) and (5) hold by proving that

$$\langle \boldsymbol{b}_L, \boldsymbol{b}_R \circ \boldsymbol{y}^{n_1} \rangle = -\langle \boldsymbol{\gamma}, \boldsymbol{y}^{n_1} \rangle \text{ and } \langle \boldsymbol{\alpha} \circ \boldsymbol{b}_L - \boldsymbol{b}_R + \boldsymbol{\beta}, \boldsymbol{y}^{n_1} \rangle = 0. \tag{8}$$

Then, one can prove that (2) holds by proving that (8) and $\phi_i(\boldsymbol{b}_L) = d_i$ hold for $1 \le i \le k$. We can combine these equations into one using the same technique: the verifier chooses $z \leftarrow_{\$} \mathbb{Z}_p$, and then the prover proves that

$$\begin{aligned} \sum_{i=1}^{k} z^{i+1} \cdot \langle \boldsymbol{b}_L, \boldsymbol{\zeta}_i \rangle + z \cdot \langle \boldsymbol{\alpha} \circ \boldsymbol{b}_L - \boldsymbol{b}_R + \boldsymbol{\beta}, \boldsymbol{y}^{n_1} \rangle + \langle \boldsymbol{b}_L, \boldsymbol{b_R} \circ \boldsymbol{y}^{n_1} \rangle \\ = \sum_{i=1}^{k} z^{i+1} \cdot d_i - \langle \boldsymbol{\gamma}, \boldsymbol{y}^{n_1} \rangle. \end{aligned} \tag{9}$$

This equality can be re-written as:

$$\langle \boldsymbol{b}_L - z\cdot\mathbf{1}^{n_1}, \boldsymbol{y}^{n_1}\circ(\boldsymbol{b}_R + z\cdot\boldsymbol{\alpha}) + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta}_i\rangle = \delta(y,z), \quad (10)$$

where $\delta(y,z) = -\sum_{i=1}^{k} z^{i+2}\cdot\langle\boldsymbol{\zeta}_i, \mathbf{1}^{n_1}\rangle + \sum_{i=1}^{k} z^{i+1}\cdot d_i - z^2\cdot\langle\boldsymbol{\alpha}, \boldsymbol{y}^{n_1}\rangle - z\cdot\langle\boldsymbol{\beta}, \boldsymbol{y}^{n_1}\rangle - \langle\boldsymbol{\gamma}, \boldsymbol{y}^{n_1}\rangle$ is a quantity that the verifier can easily calculate, the problem of proving that (2) holds is reduced to proving one single inner-product equality.

We show the full protocol $\tilde{\Pi}_{\texttt{inner}}$ between the prover and the verifier in Protocol 1, and it consists of five moves: the first two moves between the prover and the verifier are used to commit $\boldsymbol{s}_L$, $\boldsymbol{s}_M$, and $\boldsymbol{s}_R$, which is used to blind $\boldsymbol{b}_L$, $\boldsymbol{a}$, and $\boldsymbol{b}_R$, respectively.

With y and z, we define two linear vector polynomials $l(X), r(X) \in \mathbb{Z}_p^{n_1}[X]$, which are the blinded vectors of the inner-product in (10) using blinding vectors $\boldsymbol{s}_L$ and $\boldsymbol{s}_R$. Due to $\boldsymbol{s}_L$ and $\boldsymbol{s}_R$, one can publish $l(x)$ and $r(x)$ for $x \leftarrow_{\$} \mathbb{Z}_p$ without revealing any information about $\boldsymbol{b}_L$ and $\boldsymbol{b}_R$. Meanwhile, we define the inner-product polynomial $t(X) = \langle l(X), r(X)\rangle$ as [14], and the constant term of $t(x)$, denoted as t_0, is the result of the inner-product in (10). The representations of $l(X)$, $r(X)$, and $t(X)$ are defined as follows: $l(X) = \boldsymbol{b}_L - z\cdot\mathbf{1}^{n_1} + s_L\cdot X$, $r(X) = \boldsymbol{y}^{n_1}\circ(\boldsymbol{b}_R + z\cdot\boldsymbol{\alpha} + \boldsymbol{s}_R\cdot X) + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta_i}$, $t(X) = \langle l(X), r(X)\rangle = t_0 + t_1\cdot X + t_2\cdot X^2$.

Now (10) is equal to $t_0 = \delta(y,z)$, i.e., the prover needs to convince the verifier that $t_0 = -\sum_{i=1}^{k} z^{i+2}\cdot\langle\boldsymbol{\zeta}_i, \mathbf{1}^{n_1}\rangle + \sum_{i=1}^{k} z^{i+1}\cdot d_i - z^2\cdot\langle\boldsymbol{\alpha}, \boldsymbol{y}^{n_1}\rangle - z\cdot\langle\boldsymbol{\beta}, \boldsymbol{y}^{n_1}\rangle - \langle\boldsymbol{\gamma}, \boldsymbol{y}^{n_1}\rangle$ holds.

The last three moves between the prover and the verifier are used to commit coefficients t_1 and t_2 of $t(X)$ and then reveal the value of $t(X)$ at a random point x chosen by the verifier. After that, the protocol shows that the revealed value is consistent with all the former commitments.

During the verification procedure, the vector of generators $\boldsymbol{h} = (h_1, \cdots, h_{n_1})$ is switched to $\boldsymbol{h'} = \boldsymbol{h}^{\circ\boldsymbol{y}^{-n_1}} = (h_1, h_2^{(y^{-1})}, \cdots, h_{n_1}^{(y^{-n_1+1})})$ to construct a commitment to $\boldsymbol{y}^{n_1}\circ\boldsymbol{b}_R$ for $r(X)$. Hence, $\tilde{P}$ and S is the vector commitment of $(\boldsymbol{b}_L, \boldsymbol{y}^{n_1}\circ\boldsymbol{b}_R, \boldsymbol{a})$ and $(\boldsymbol{s}_L, \boldsymbol{y}^{n_1}\circ\boldsymbol{s}_R, \boldsymbol{s}_M)$, respectively.

The first verification equation of Protocol 1 is to check whether equation (9) is consistent with $\boldsymbol{b}_L$, i.e., check the constant term of $t(X)$, $t_0 \stackrel{?}{=} \delta(y,z)$, at a random point x. The remaining verification equations check whether the commitments used during the protocol are consistent.

Since the prover only needs to prove the knowledge of $\boldsymbol{a}$, a three-move Σ-protocol [42] is used inside $\tilde{\Pi}_{\texttt{inner}}$ with S as the first-move commitment from the prover, x as the challenge from the verifier, and $\boldsymbol{\eta}$ as the third-move response from the prover.

We present the security of $\tilde{\Pi}_{\texttt{inner}}$ as an interactive protocol in Theorem 1 of Section 3.3.

$\tilde{\Pi}_{\texttt{inner}}\langle\mathcal{P}(\boldsymbol{g}, \boldsymbol{g}_2, \boldsymbol{h}, h, d_i, \tilde{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \phi_i; \boldsymbol{b}_L, \boldsymbol{b}_R, \boldsymbol{a}, \rho_L),$
$\mathcal{V}(\boldsymbol{g}, \boldsymbol{g}_2, \boldsymbol{h}, h, d_i, \tilde{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \phi_i)\rangle$ where $1 \le i \le k$

$\mathcal{P}$ computes :
$\boldsymbol{s}_L, \boldsymbol{s}_R \leftarrow\$ \mathbb{Z}_p^{n_1}, \boldsymbol{s}_M \leftarrow\$ \mathbb{Z}_p^{n_2}, \rho_S \leftarrow\$ \mathbb{Z}_p$
$S = h^{\rho_S}\cdot\boldsymbol{g}^{\boldsymbol{s}_L}\cdot\boldsymbol{g}_2^{\boldsymbol{s}_M}\cdot\boldsymbol{h}^{\boldsymbol{s}_R} \in \mathbb{G}$
$\mathcal{P} \to \mathcal{V} : S$
$\mathcal{V}$ computes :
$y, z \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \to \mathcal{P} : y, z$
$\mathcal{P}$ computes:
$l(X) = \boldsymbol{b}_L - z\cdot\mathbf{1}^{n_1} + \boldsymbol{s}_L\cdot X$
$r(X) = \boldsymbol{y}^{n_1}\circ(\boldsymbol{b}_R + z\cdot\boldsymbol{\alpha} + \boldsymbol{s}_R\cdot X) + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta_i}$
$t(X) = \langle l(X), r(X)\rangle = t_0 + t_1 X + t_2 X^2$
$\tau_1, \tau_2 \leftarrow\$ \mathbb{Z}_p, T_1 = g^{t_1}h^{\tau_1}, T_2 = g^{t_2}h^{\tau_2}$
$\mathcal{P} \to \mathcal{V} : T_1, T_2$
$\mathcal{V}$ computes :
$x \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \to \mathcal{P} : x$
$\mathcal{P}$ computes:
$\boldsymbol{l} = l(x) = \boldsymbol{b}_L - z\cdot\mathbf{1}^{n_1} + x\cdot\boldsymbol{s}_L$
$\boldsymbol{r} = r(x) = \boldsymbol{y}^{n_1}\circ(\boldsymbol{b}_R + z\cdot\boldsymbol{\alpha} + x\cdot\boldsymbol{s}_R) + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta_i}$
$\boldsymbol{\eta} = \boldsymbol{a} + x\cdot\boldsymbol{s}_M$
$\hat{t} = \langle\boldsymbol{l}, \boldsymbol{r}\rangle$
$\tau_x = \tau_2\cdot x^2 + \tau_1\cdot x, \quad \mu = \rho_L + x\cdot\rho_S$
$\mathcal{P} \to \mathcal{V} : \tau_x, \mu, \hat{t}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{\eta}$
$\mathcal{V}$: Verification procedure:
Let $\boldsymbol{h'} = \boldsymbol{h}^{\circ\boldsymbol{y}^{-n_1}}$,
$g^{\hat{t}-\delta(y,z)}h^{\tau_x} \stackrel{?}{=} T_1^x\cdot T_2^{x^2}$
Let $T = \tilde{P}\cdot S^x\cdot\boldsymbol{g}^{-z\cdot\mathbf{1}^{n_1}}\cdot(\boldsymbol{h'})^{z\cdot\boldsymbol{\alpha}\circ\boldsymbol{y}^{n_1} + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta}_i}$
$T \stackrel{?}{=} h^{\mu}\cdot\boldsymbol{g}^{\boldsymbol{l}}\cdot(\boldsymbol{h'})^{\boldsymbol{r}}\cdot\boldsymbol{g}_2^{\boldsymbol{\eta}}$
$\hat{t} \stackrel{?}{=} \langle\boldsymbol{l}, \boldsymbol{r}\rangle$

Protocol 1: The $\tilde{\Pi}_{\texttt{inner}}$ protocol.

3.2. The Outer Protocol

In Section 3.1, we have constructed the protocol to prove that secret vectors $\boldsymbol{b}_L \in \mathbb{Z}_p^{n_1}$ and $\boldsymbol{a} \in \mathbb{Z}_p^{n_2}$ satisfy relation $\mathcal{R}_{\texttt{inner}}$ without leaking any information about $\boldsymbol{b}_L$ and $\boldsymbol{a}$. The information about $\boldsymbol{b}_L$ is hidden in the Pedersen vector commitment $P = h^{\rho_L}\cdot\boldsymbol{g}_1^{\boldsymbol{b}_L}\cdot\boldsymbol{g}_2^{\boldsymbol{a}}$, and the DL relations among $\boldsymbol{g}_1, \boldsymbol{g}_2$ and h are unknown. However, in some circumstances, the prover may know the DL relations among the generators, and the discrete logarithm relation assumption (Definition 2) no longer holds, making the soundness proofs of Bulletproofs-like protocols infeasible. In the following part of this section, we will give the `outer` protocol $\Pi_{\texttt{KKDL}}$,

which proves $\mathcal{R}_{\texttt{KKDL}}$ of (1) using the `inner` protocol as a sub-protocol.

Following the basic ideas of [22], we give an SHVZK protocol for the relation $\mathcal{R}_{\texttt{KKDL}}$ by combining the equality $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ of (1) with the commitment $P = h^{\rho_L} \cdot \hat{\boldsymbol{g}}_1^{\boldsymbol{b}_L} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{a}}$ as follows, where h, $\hat{\boldsymbol{g}}_1$, and $\hat{\boldsymbol{g}}_2$ are chosen randomly by the verifier at the beginning of $\Pi_{\texttt{KKDL}}$ [4].

1) Introduce two new random challenges, e and v, from the verifier. The challenge v is used to gather all equalities $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ into one equality $\prod_{i=1}^{m} (\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i})^{v^{i-1}} = (\prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{v^{i-1}})^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\sum_{i=1}^{m} v^{i-1} \cdot \boldsymbol{\psi}_i} = 1$. Let $\boldsymbol{a} = \sum_{i=1}^{m} v^{i-1} \cdot \boldsymbol{\psi}_i$, the challenge e is used to combine the gathered equality with the commitment as follows:

$$\begin{aligned} P &= h^{\rho_L} \cdot \hat{\boldsymbol{g}}_1^{\boldsymbol{b}_L} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{a}} \cdot [(\prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{v^{i-1}})^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{a}}]^e \\ &= h^{\rho_L} \cdot (\hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}})^{\boldsymbol{b}_L} \cdot (\hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^{e})^{\boldsymbol{a}} \end{aligned} \tag{11}$$

Since the DL relations among h, $\hat{\boldsymbol{g}}_1$, and $\hat{\boldsymbol{g}}_2$ are unknown, the DL relations among h, $\boldsymbol{g}_1$, and $\boldsymbol{g}_2$ are unknown, where $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are defined as follows:

$$\boldsymbol{g}_1 = \hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}} \text{ and } \boldsymbol{g}_2 = \hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^{e} \tag{12}$$

Therefore, we can use $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ of (12) and $(d_1, \cdots, d_k, h, P, f, \phi_1, \cdots, \phi_k)$ of $\mathcal{R}_{\texttt{KKDL}}$ as the public inputs of $\mathcal{R}_{\texttt{inner}}$ of Section 3.1, and use $(\boldsymbol{b}_L, \boldsymbol{a}, \rho_L)$ as the witnesses of $\mathcal{R}_{\texttt{inner}}$, and then execute the SHVZK protocol of Section 3.1.

2) In [22], randomness $v \in \mathbb{Z}_p$ is sent at the beginning of the protocol before committing $\boldsymbol{b}_L$, which will bring the following two problems:
   a) Since v is sent before committing $\boldsymbol{b}_L$, the prover may choose $\boldsymbol{b}_L$ depending on v, which will make the commitment P unfixed and make the witness $\boldsymbol{\psi}_i$ unextractable in the soundness proof. More concretely, one can extract $(\boldsymbol{b}_L, \boldsymbol{a}, \rho_L)$ from the `inner` protocol satisfying

$$\begin{aligned} P &= h^{\rho_L} \cdot \boldsymbol{g}_1^{\boldsymbol{b}_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \\ &= h^{\rho_L} \cdot (\hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}})^{\boldsymbol{b}_L} \cdot (\hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^{e})^{\boldsymbol{a}}. \end{aligned} \tag{13}$$

   Then, one can deduce equality $(\prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{v^{i-1}})^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{a}} = 1$ by fixing v. For m different values of v, one can deduce m such equalities by the rewinding technique. However, since $\boldsymbol{b}_L$ may be different, one cannot deduce $\boldsymbol{\psi}_i$ that satisfies $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ for $1 \le i \le m$ with the same $\boldsymbol{b}_L$.
   b) Since v is sent at the beginning of the protocol, one cannot rewind v many times using the Fiat-Shamir heuristic [32] because all the protocol statements are fixed. The authors of [22] use the Schwartz-Zippel lemma [26] to overcome this. For the Schwartz-Zippel lemma, the polynomial coefficients should be fixed before choosing the random variable (i.e., v). However, when they apply the Schwartz-Zippel lemma, the coefficients may be selected according to the random variable since v is sent at the beginning of the protocol, which will make the soundness proof infeasible.

   To overcome the problems mentioned above, since P is the commitment of $(\boldsymbol{a}, \boldsymbol{b}_L)$ and $\boldsymbol{a}$ is based on v, we split the commitment into two parts. Firstly, the prover commits $\boldsymbol{b}_L$ at the beginning of the protocol before receiving v. Secondly, the prover commits $\boldsymbol{a}$ after receiving v and adds an in-line Σ-protocol to ensure that this commitment does not contain any information about $\boldsymbol{b}_L$. Then, P is the combination of these two commitments. These methods solve Problem (a) since $\boldsymbol{b}_L$ is fixed at the beginning of the protocol and solve Problem (b) since one can rewind v many times in the soundness proof for the reason that v is chosen after committing $\boldsymbol{b}_L$. We show the detailed explanation in the witness-extended emulation part of the proof of Theorem 2.

We give the SHVZK protocol $\Pi_{\texttt{KKDL}}$ for the relation $\mathcal{R}_{\texttt{KKDL}}$ in Protocol 2, consisting of six moves. Firstly, the prover commits $\boldsymbol{b}_L$ and sends the commitment P_1 to the verifier. Then, the verifier chooses $v \leftarrow_{\$} \mathbb{Z}_p$ and sends v to the prover. After receiving the challenge v from the verifier, the prover and the verifier conduct the three-move in-line Σ-protocol using $\boldsymbol{a} = \sum_{i=1}^{m} v^{i-1} \cdot \boldsymbol{a}_i \in \mathbb{Z}_p^{n_2}$ as the witness, where the commitment of $\boldsymbol{a}$ is denoted as P_2. Here, we use $P = P_1 \cdot P_2$ as the commitment of $\boldsymbol{b}_L$ and $\boldsymbol{a}$. Afterward, the verifier chooses $e \leftarrow_{\$} \mathbb{Z}_p$ and sends e to the prover. Finally, the prover and the verifier proceed to the protocol for $\mathcal{R}_{\texttt{inner}}$ of Section 3.1 using $(\boldsymbol{g}_1 \leftarrow \hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}}, \boldsymbol{g}_2 \leftarrow \hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^{e}, d_1, \cdots, d_k, h, P, f, \phi_1, \cdots, \phi_k)$ as the public inputs and $(\boldsymbol{b}_L, \boldsymbol{a}, \rho_L)$ as the witnesses.

We show the security of $\Pi_{\texttt{KKDL}}$ as an interactive protocol in Theorem 2 of Section 3.3. Since the verifier is a public-coin verifier, we can convert the protocol into a non-interactive protocol that is secure and full zero-knowledge in the random oracle model using the Fiat-Shamir heuristic [32].

3.3. Security Analysis

In this section, we prove the security of $\Pi_{\texttt{inner}}$ and $\Pi_{\texttt{KKDL}}$ presented in Section 3.1 and Section 3.2, respectively. We give the security analysis of $\Pi_{\texttt{inner}}$ in Theorem 1 and the security analysis of $\Pi_{\texttt{KKDL}}$ in Theorem 2.

Theorem 1 (Security of $\Pi_{\texttt{inner}}$)**.** *The protocol $\Pi_{\texttt{inner}}$ presented in Section 3.1 has perfect completeness, perfect*

[4] In the non-interactive version, h, $\hat{\boldsymbol{g}}_1$, and $\hat{\boldsymbol{g}}_2$ can be deduced by inputting $\boldsymbol{q}_{1,i}$ and $\boldsymbol{q}_2$ to some cryptographic-secure hash-to-point functions [21], which will make the DL relations between $(\boldsymbol{q}_{1,i}, \boldsymbol{q}_2)$ and $(h, \hat{\boldsymbol{g}}_1, \hat{\boldsymbol{g}}_2)$ unknown.

$\Pi_{\texttt{KKDL}} : \langle \mathcal{P}(d_i, \phi_i, \boldsymbol{q}_{1,j}, \boldsymbol{q}_2, f; \boldsymbol{b}_L, \boldsymbol{\psi}_j), \mathcal{V}(d_i, \phi_i, \boldsymbol{q}_{1,j}, \boldsymbol{q}_2, f)\rangle$
where $1 \le i \le k$ and $1 \le j \le m$

$\mathcal{V}$ computes:
$h \leftarrow\$ \mathbb{G}, \hat{\boldsymbol{g}}_1 \leftarrow\$ \mathbb{G}^{n_1}$,
$\hat{\boldsymbol{g}}_2 \leftarrow\$ \mathbb{G}^{n_2}, \boldsymbol{h} \leftarrow\$ \mathbb{G}^{n_1}$
$\mathcal{V} \rightarrow \mathcal{P} : h, \hat{\boldsymbol{g}}_1, \hat{\boldsymbol{g}}_2, \boldsymbol{h}$
$\mathcal{P}$ computes:
$\rho_1 \leftarrow\$ \mathbb{Z}_p, P_1 = h^{\rho_1} \cdot \hat{\boldsymbol{g}}_1^{\boldsymbol{b}_L}$
$\mathcal{P} \rightarrow \mathcal{V} : P_1$
$\mathcal{V}$ computes:
$v \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : v$

..... In-line Σ-protocol .....

$\mathcal{P}$ computes:
Let $\boldsymbol{a} = \sum_{i=1}^{m} v^{i-1} \cdot \boldsymbol{\psi}_i \in \mathbb{Z}_p^{n_2}$,
$\rho_2, \rho_3 \leftarrow\$ \mathbb{Z}_p, \boldsymbol{c} \leftarrow\$ \mathbb{Z}_p^{n_2}$,
$P_2 = h^{\rho_2} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{a}}, P_3 = h^{\rho_3} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{c}}$
$\mathcal{P} \rightarrow \mathcal{V} : P_2, P_3$
$\mathcal{V}$ computes:
$w \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : w$
$\mathcal{P}$ computes:
$\theta_1 = \rho_3 + w \cdot \rho_2, \boldsymbol{\theta}_2 = \boldsymbol{c} + w \cdot \boldsymbol{a}$
$\mathcal{P} \rightarrow \mathcal{V} : \theta_1, \boldsymbol{\theta}_2$
$\mathcal{V}$: Verification Procedure:
$h^{\theta_1} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{\theta}_2} \stackrel{?}{=} P_3 \cdot P_2^w$

If passed, both prover and verifier compute $P = P_1 \cdot P_2$
$\mathcal{V}$ computes:
$e \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : e$
Proceed to the proof of $\mathcal{R}_{\texttt{inner}}$ using $\boldsymbol{g}_1 \leftarrow \hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}}$, $\boldsymbol{g}_2 \leftarrow \hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^e, d_1, \cdots, d_k, h, P, f, \phi_1, \cdots, \phi_k)$ as public inputs, and $(\boldsymbol{b}_L, \boldsymbol{a}, \rho_L = \rho_1 + \rho_2)$ as witnesses of prover.

Protocol 2: The $\Pi_{\texttt{KKDL}}$ protocol.

special honest-verifier zero-knowledge, and computational witness-extended emulation.

The proof process follows the process of Bulletproofs, we show the detailed proof process in Appendix C.

Theorem 2 (Security of $\Pi_{\texttt{KKDL}}$)**.** *The protocol Π_{KKDL} presented in Section 3.2 has perfect completeness, perfect special honest-verifier zero-knowledge, and computational witness-extended emulation.*

Proof. 1) **Perfect completeness.** Perfect completeness follows directly.

2) **Perfect special honest-verifier zero-knowledge.** The proof of special honest-verifier zero-knowledge is almost the same as the proof of Theorem 1 except (1) the randomness set is (v, w, e, y, z, x); (2) the commitment $P = \tilde{P} \cdot \boldsymbol{h}^{-\boldsymbol{\beta}}$ is chosen at random; (3) the generator $\boldsymbol{g} = \boldsymbol{g}_1 \circ \boldsymbol{h}^{\circ -\boldsymbol{\alpha}} = \hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}} \circ \boldsymbol{h}^{\circ -\boldsymbol{\alpha}}$ and $\boldsymbol{g}_2 = \hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^e$; (4) the special honest-verifier zero-knowledge of the in-line Σ-protocol can be deduced from the special honest-verifier zero-knowledge property of Σ-protocol [42] directly.

3) **Computational witness-extended emulation.** We construct an extractor $\chi_{\texttt{KKDL}}$ to prove the computational witness-extended emulation. The extractor runs the prover with m different values of v, two different values of w, and two different values of e. Additionally, it invokes the extractor $\chi_{\texttt{inner}}$ for the relation $\mathcal{R}_{\texttt{inner}}$ of Theorem 1 on each of (v, w, e), resulting in $4m$ total transcripts.
For each (v, w, e), the extractor $\chi_{\texttt{KKDL}}$ first runs the extractor $\chi_{\texttt{inner}}$ to extract a witness $(\boldsymbol{b}_L, \boldsymbol{a}, \rho_L)$ for the relation $\mathcal{R}_{\texttt{inner}}$, satisfying

$$\begin{aligned} P &= h^{\rho_L} \cdot \boldsymbol{g}_1^{\boldsymbol{b}_L} \cdot \boldsymbol{g}_2^{\boldsymbol{a}} \\ &= h^{\rho_L} \cdot (\hat{\boldsymbol{g}}_1 \circ \prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{ev^{i-1}})^{\boldsymbol{b}_L} \cdot (\hat{\boldsymbol{g}}_2 \circ \boldsymbol{q}_2^e)^{\boldsymbol{a}} \end{aligned} \tag{14}$$

For given values of v, using two valid transcripts for different e challenges e and e', one has that

$$h^{\rho_L - \rho'_L} \cdot \hat{\boldsymbol{g}}_1^{\boldsymbol{b}_L - \boldsymbol{b}'_L} \cdot (\prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{v^{i-1}})^{e \cdot \boldsymbol{b}_L - e' \cdot \boldsymbol{b}'_L} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{a} - \boldsymbol{a}'} \cdot \boldsymbol{q}_2^{e \cdot \boldsymbol{a} - e' \cdot \boldsymbol{a}'} = 1, \tag{15}$$

where $(\boldsymbol{b}_L, \boldsymbol{a}, \rho_L)$ and $(\boldsymbol{b}'_L, \boldsymbol{a}', \rho'_L)$ is the transcript for e and e', respectively. The correctness of this step is based on the general forking lemma [16] which is also used in Bulletproofs and Omniring. The commitments and randomnesses form a tree of accepting transcripts, where the nodes denote the commitments and the edges denote the randomness. One parent node has different child nodes (subtrees) w.r.t. different randomnesses, but the parent node is fixed for its subtrees, and one can extract the witnesses of this parent node from different transcripts of its subtree using the rewinding technique. Meanwhile, the parent node can be used to extract the grandparent node with its peers. Hence, the commitment P is fixed for its subtrees starting from e and e', so one can deduce two different transcripts satisfying (14) with the same P.
From (15), one can deduce that $\rho_L = \rho'_L$, $\boldsymbol{b}_L = \boldsymbol{b}'_L$ and $\boldsymbol{a} = \boldsymbol{a}'$. Otherwise, we can deduce a non-trivial DL

relation among generators h, $\hat{\boldsymbol{g}}_1$, and $\hat{\boldsymbol{g}}_2$, contradicting the DL relation assumption. Then, we have

$$(\prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{v^{i-1}})^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{a}} = 1 \tag{16}$$

Next, let's turn to the in-line Σ-protocol. Using two valid transcripts $(P_2, P_3, \theta_1, \boldsymbol{\theta}_2)$ and $(P_2', P_3', \theta_1', \boldsymbol{\theta}_2')$ for different w challenges w and w', one has that

$$P_2 = h^{\frac{\theta_1 - \theta_1'}{w - w'}} \cdot \hat{\boldsymbol{g}}_2^{\frac{\boldsymbol{\theta}_2 - \boldsymbol{\theta}_2'}{w - w'}} \text{ and } P_3 = h^{\frac{w \cdot \theta_1' - w' \cdot \theta_1}{w - w'}} \cdot \hat{\boldsymbol{g}}_2^{\frac{w \cdot \boldsymbol{\theta}_2' - w' \cdot \boldsymbol{\theta}_2}{w - w'}} \tag{17}$$

Otherwise, one can deduce a non-trivial DL relation among generators h and $\hat{\boldsymbol{g}}_2$.

By (14) and (17), one can deduce that $\hat{\boldsymbol{g}}_1^{\boldsymbol{b}_L}$ of P has nothing to do with P_2, i.e., it all comes from P_1. Therefore, we have that $\boldsymbol{b}_L$ is fixed and has nothing to do with v by the binding property of the commitment scheme. Then, one can deduce that $\boldsymbol{b}_L$ of (16) is fixed for different values of v.

Using m valid transcripts $\boldsymbol{a}_i$ for different v challenges v_i where $1 \leq i \leq m$, constructing a system of equations for unknown variables $\boldsymbol{\psi}_1 \in \mathbb{Z}_p^{n_2}, \cdots, \boldsymbol{\psi}_m \in \mathbb{Z}_p^{n_2}$ as follows:

$$\begin{cases} \sum_{i=1}^{m} v_1^i \cdot \boldsymbol{\psi}_i = \boldsymbol{a}_1 \\ \cdots \\ \sum_{i=1}^{m} v_m^i \cdot \boldsymbol{\psi}_i = \boldsymbol{a}_m \end{cases} \tag{18}$$

Since the coefficient matrix of (18) is a Vandermonde matrix, there exists one solution $(\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_m)$ for (18). Then, one can deduce the following equality

$$\prod_{i=1}^{m} (\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i})^{v^{i-1} \cdot} = (\prod_{i=1}^{m} \circ \boldsymbol{q}_{1,i}^{v^{i-1}})^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\sum_{i=1}^{m} v^{i-1} \cdot \boldsymbol{\psi}_i} = 1. \tag{19}$$

Since (19) holds for m different values of v, we can deduce that $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ for $1 \leq i \leq m$. Of course, one can sample fresh $\psi_1, \cdots, \psi_m$ depending on v_i, but the verifier will not see this, the only thing they can see is $\boldsymbol{a} = \sum_{i=1}^{m} v^{i-1} \cdot \psi_i$ and the final verification processes are passed. We can treat different $(\psi_1, \cdots, \psi_m)$ as the same and construct the system of equations. Since the coefficient matrix is a Vandermonde matrix, one can always get the only value of $(\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_m)$ satisfying $\prod_{i=1}^{m} (\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i})^{v^{i-1} \cdot} = 1$, i.e., $\boldsymbol{q}_{1,i}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_i} = 1$ for $1 \leq i \leq m$. These equations are sufficient for our $R_{\texttt{KKDL}}$ since we only consider the relations. Therefore, one can deduce $\mathcal{R}_{\texttt{KKDL}}$ of (1).

□

3.4. Improved Inner-Product Argument and Logarithmic-Size Proofs

3.4.1. Improved Inner-Product Argument. The maximum component of Bulletproofs is two N-dimensional vectors $\boldsymbol{l}$ and $\boldsymbol{r}$ satisfying relation (20) where $\boldsymbol{g}, \boldsymbol{h} \in \mathbb{G}^N$, $P \in \mathbb{G}$, $\hat{t} \in \mathbb{Z}_p$ and $\boldsymbol{l}, \boldsymbol{r} \in \mathbb{Z}_p^N$. Bulletproofs employs the improved inner-product argument to reduce the size of $(\boldsymbol{l}, \boldsymbol{r})$ that satisfies the following relation from $2N$ to $2\lceil \log_2(N) \rceil + 2$:

$$\mathcal{R}_{\texttt{bullet}} = \{(\boldsymbol{g}, \boldsymbol{h}, P, \hat{t}) | \exists\ \boldsymbol{l}, \boldsymbol{r}, \text{ s.t., } P = \boldsymbol{g}^{\boldsymbol{l}} \boldsymbol{h}^{\boldsymbol{r}} \wedge \hat{t} = \langle \boldsymbol{l}, \boldsymbol{r} \rangle\}. \tag{20}$$

In [17], Attema and Cramer generalize the improved inner-product argument, and give an argument of knowledge for the following relation (Section 4 of [17]).

$$\mathcal{R}_{\texttt{ac20}} = \{(\tilde{\boldsymbol{g}}, P, L, \hat{t}) \mid \exists\ \boldsymbol{z}, , \text{ s.t., } P = \tilde{\boldsymbol{g}}^{\boldsymbol{z}} \wedge \hat{t} = L(\boldsymbol{z})\}, \tag{21}$$

where $\tilde{\boldsymbol{g}} \in \mathbb{G}^N$, $P \in \mathbb{G}$, $\boldsymbol{z} \in \mathbb{Z}_p^N$ and $L : \mathbb{Z}_q^N \to \mathbb{Z}_q$ is a linear function. The size of this argument of knowledge is $2\lceil \log_2(N) \rceil$. The basic idea of the protocol for $\mathcal{R}_{\texttt{ac20}}$ is almost the same as $\mathcal{R}_{\texttt{bullet}}$ by representing $P = \tilde{\boldsymbol{g}}_L^{\boldsymbol{z}_L} \cdot \tilde{\boldsymbol{g}}_R^{\boldsymbol{z}_R}$, where the dimension of $\tilde{\boldsymbol{g}}_L$, $\tilde{\boldsymbol{g}}_R$, $\boldsymbol{z}_L$ and $\boldsymbol{z}_R$ is $\lceil \frac{N}{2} \rceil$.

3.4.2. Logarithmic Size Proofs. Using the Fiat-Shamir heuristic, $\Pi_{\texttt{inner}}$ consists of three elements $S, T_1, T_2 \in \mathbb{G}$, three elements $\tau_x, \mu, \hat{t} \in \mathbb{Z}_p$, two n_1-dimensional vectors $\boldsymbol{l}, \boldsymbol{r} \in \mathbb{Z}_p^{n_1}$, and one n_2-dimensional vector $\boldsymbol{\eta} \in \mathbb{Z}_p^{n_2}$. Hence, the total proof size is $2n_1 + n_2 + 6$.

The maximum component of $\Pi_{\texttt{inner}}$ is two n_1-dimensional vectors $\boldsymbol{l}, \boldsymbol{r}$ and one n_2-dimensional vector $\boldsymbol{\eta}$ satisfying the following relation from Protocol 1:

$$\mathcal{R}_{\texttt{inner-ipa}} = \left\{ (\boldsymbol{g}, \boldsymbol{h}', \boldsymbol{g}_2, P, \hat{t}) \;\middle|\; \begin{array}{c} \exists\ \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{\eta}, \text{ s.t., } \hat{t} = \langle \boldsymbol{l}, \boldsymbol{r} \rangle \\ \wedge\ P = \boldsymbol{g}^{\boldsymbol{l}} \cdot (\boldsymbol{h}')^{\boldsymbol{r}} \cdot \boldsymbol{g}_2^{\boldsymbol{\eta}} \end{array} \right\}, \tag{22}$$

where $\boldsymbol{g}, \boldsymbol{h}' \in \mathbb{G}^{n_1}$ and $\boldsymbol{g}_2 \in \mathbb{G}^{n_2}$. Let $\tilde{\boldsymbol{g}} = (\boldsymbol{g}, \tilde{\boldsymbol{g}}_1, \boldsymbol{h}', \tilde{\boldsymbol{g}}_2)$, $\boldsymbol{z} = (\boldsymbol{l}, \boldsymbol{\eta}_1, \boldsymbol{r}, \boldsymbol{\eta}_2)$, $N = 2n_1 + n_2$ and $L(\boldsymbol{z}) = \langle \boldsymbol{l}, \boldsymbol{r} \rangle$, and then one can transform relation (22) to relation (21) and use the PoK in Section 4 of [17] to compress the size of $(\boldsymbol{l}, \boldsymbol{r}, \boldsymbol{\eta})$ from $2n_1 + n_2$ to $2\lceil \log_2(2n_1 + n_2) \rceil$, where $(\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2) = \boldsymbol{g}_2$, $(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) = \boldsymbol{z}$, $\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2 \in \mathbb{G}^{\lceil \frac{n_2}{2} \rceil}$ and $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \mathbb{Z}_p^{\lceil \frac{n_2}{2} \rceil}$.

Using the Fiat-Shamir heuristic, $\Pi_{\texttt{KKDL}}$ consists of three elements $P_1, P_2, P_3 \in \mathbb{G}$, one element $\theta_1 \in \mathbb{Z}_p$, and one n_2-dimensional vector $\boldsymbol{\theta_2} \in \mathbb{Z}_p^{n_2}$. Hence, the total proof size is $n_2 + 4$.

The maximum component of $\Pi_{\texttt{KKDL}}$ is one n_2-dimensional vector $\boldsymbol{\theta}_2$ and one element θ_1 satisfying the following relation from Protocol 2

$$\mathcal{R}_{\texttt{outer-ipa}} = \{(h, \hat{\boldsymbol{g}}_2, P) \mid \exists\ \theta_1, \boldsymbol{\theta}_2, \text{ s.t., } P = h^{\theta_1} \cdot \hat{\boldsymbol{g}}_2^{\boldsymbol{\theta}_2}\}, \tag{23}$$

where $\hat{\boldsymbol{g}}_2 \in \mathbb{G}^{n_2}$ and $h \in \mathbb{G}$. Let $\tilde{\boldsymbol{g}} = (h, \hat{\boldsymbol{g}}_2)$, $\boldsymbol{z} = (\theta_1, \boldsymbol{\theta}_2)$, $N = n_2 + 1$ and $L(\boldsymbol{z}) = 0$, and then one can transform relation (23) to relation (21) and use the PoK in Section 4 of [17] to compress the size of $(\theta_1, \boldsymbol{\theta}_2)$ from $n_2 + 1$ to $2\lceil \log_2(n_2 + 1) \rceil$.

Combining $\Pi_{\texttt{inner}}$ and $\Pi_{\texttt{KKDL}}$, the proof size of the whole protocol is $2 \cdot \lceil \log_2(2n_1 + n_2) \rceil + 2 \cdot \lceil \log_2(n_2 + 1) \rceil + 9$.

We measure the time complexity using the number of exponentiations. For the proving time, $\Pi_{\texttt{inner}}$ requires about

$13n_1 + 5n_2$ exponentiations (including $8n_1 + 4n_2$ exponentiations for inner-product argument), $\Pi_{\texttt{inner}}$ requires about $n_1 + 6n_2$ exponentiations (including $4n_2$ exponentiations for inner-product argument), and transforming the parameters from $\Pi_{\texttt{inner}}$ to $\Pi_{\texttt{outer}}$ requires about $m \cdot n_1 + n_2$ exponentiations. Hence, the proving time is $(m+14)n_1 + 12n_2$ exponentiations. Using the technique of [15], the verification time is managed by a single multi-exponentiation of size about $4n_1 + 4n_2 + \lceil \log_2(2n_1 + n_2) \rceil + \lceil \log_2(n_2 + 1) \rceil$.

4. Application: Omniring-style RingCT

In this section, we give a RingCT construction in the Omniring framework [22] which doesn't have the problems mentioned in Section 1.1.3 and 3.2. We use the same notations as [22] in this section. Let $\boldsymbol{G}' = (\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \boldsymbol{g}_4, \boldsymbol{g}_5) \leftarrow_{\$} \mathbb{G}^{m-|\mathcal{R}|-3}$ where $\boldsymbol{g}_1 \in \mathbb{G}^{|\mathcal{R}||\mathcal{S}|}$, $\boldsymbol{g}_2 \in \mathbb{G}^{\beta|\mathcal{T}|}$, $\boldsymbol{g}_3 \in \mathbb{G}^{|\mathcal{S}|}$, $\boldsymbol{g}_4 \in \mathbb{G}^{|\mathcal{S}|}$, $\boldsymbol{g}_5 \in \mathbb{G}^{|\mathcal{S}|}$, and $m = 3 + |\mathcal{R}| + |\mathcal{R}||\mathcal{S}| + \beta|\mathcal{T}| + 3|\mathcal{S}|$. Also, let $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \boldsymbol{h}_4, \boldsymbol{h}_5) \leftarrow_{\$} \mathbb{G}^m$ where $\boldsymbol{h}_1 \in \mathbb{G}^{3+|\mathcal{R}|}$, $\boldsymbol{h}_2 \in \mathbb{G}^{|\mathcal{R}||\mathcal{S}|}$, $\boldsymbol{h}_3 \in \mathbb{G}^{\beta|\mathcal{T}|}$, $\boldsymbol{h}_4 \in \mathbb{G}^{2|\mathcal{S}|}$, and $\boldsymbol{h}_5 \in \mathbb{G}^{|\mathcal{S}|}$. Denote $\hat{\boldsymbol{g}}_1 = \boldsymbol{g}_1 \circ \boldsymbol{h}_2$ and $\hat{\boldsymbol{g}}_2 = \boldsymbol{g}_2 \circ \boldsymbol{h}_3$.

We replace the first two moves from Section 5.1 of [22] with an `outer` protocol shown in Protocol 3. Then, continue executing the Omniring protocol after the prover receives w from the verifier. Denote $r_A = r_{A,1} + r_{A,2}$, these steps are compatible since

$$\begin{aligned} & A_1 \cdot A_2 \cdot (\boldsymbol{h}_2 \| \boldsymbol{h}_3)^{-1} \\ =& F^{r_A} \cdot \hat{\boldsymbol{g}}_1^{\text{vec}(\mathbf{E})} \cdot \hat{\boldsymbol{g}}_2^{\text{vec}(\mathbf{B})} \cdot \boldsymbol{g}_3^{\boldsymbol{a}^{\mathcal{S}}} \cdot \boldsymbol{g}_4^{\boldsymbol{r}^{\mathcal{S}}} \cdot \boldsymbol{g}_5^{\boldsymbol{x}} \cdot \\ & \boldsymbol{h}_5^{\boldsymbol{x}^{\circ -1}} \cdot (\boldsymbol{h}_2 \| \boldsymbol{h}_3)^{-1} \cdot \boldsymbol{P}^{(\xi \| \eta \| 1 \| \hat{\boldsymbol{e}})} \\ =& F^{r_A} \cdot (\boldsymbol{P} \| \boldsymbol{G}')^{\boldsymbol{c}_L} \cdot \boldsymbol{H}^{\boldsymbol{c}_R} \\ =& A. \end{aligned} \tag{24}$$

The security proofs, including balance, privacy and non-slanderability, are inherited from [22] except for the security proofs for argument of knowledge construction (Appendix D of [22]). The completeness is quite straightforward. The zero-knowledge can be deduced from the SHVZK of Theorem 2 and the SHVZK of Omniring.

The soundness proof is almost the same except for the last few steps, i.e., steps after deducing the witnesses $(\boldsymbol{a}^{\mathcal{S}'}, \boldsymbol{r}^{\mathcal{S}'}, \boldsymbol{x}', \boldsymbol{E})$ and equations

$$\begin{aligned} \xi' &= -\langle \boldsymbol{v}^{|\mathcal{S}|}, u \cdot \boldsymbol{a}^{\mathcal{S}'} + u^2 \cdot \boldsymbol{x}'^{\circ -1} \rangle \\ \eta' &= -\langle \boldsymbol{v}^{|\mathcal{S}|}, \boldsymbol{x}' + u \cdot \boldsymbol{r}^{\mathcal{S}'} \rangle \\ \psi' &= 1 \\ \hat{\boldsymbol{e}}' &= \boldsymbol{v}^{|\mathcal{S}|} \boldsymbol{E}' = \sum_{l \in [|\mathcal{S}|]} v^{l-1} \cdot \boldsymbol{e}'_l. \end{aligned}$$

Then one can deduce

$$\begin{aligned} I = & \prod_{l \in [|\mathcal{S}|]} (H^{-x'_l} \boldsymbol{R}^{\boldsymbol{e}'_l})^{v^{l-1}} \cdot \prod_{l \in [|\mathcal{S}|]} (G^{-a_l^{\mathcal{S}'}} H^{-r_l^{\mathcal{S}'}} \boldsymbol{C}_{\mathcal{R}}^{\boldsymbol{e}'_l})^{u v^{l-1}} \cdot \\ & \prod_{l \in [|\mathcal{S}|]} (G^{\frac{-1}{x'_l}} \text{tag}_l)^{u^2 v^{l-1}}. \end{aligned} \tag{25}$$

$\Pi_{\texttt{outer}} : \langle \mathcal{P}(\boldsymbol{R}, \boldsymbol{C}_{\mathcal{R}}, \boldsymbol{T}, \boldsymbol{C}_{\mathcal{T}}; \mathbf{E}, \boldsymbol{x}, \boldsymbol{a}^{\mathcal{S}}, \boldsymbol{r}^{\mathcal{S}}, \mathbf{B}, \boldsymbol{a}^{\mathcal{T}}, \boldsymbol{r}^{\mathcal{T}}), \mathcal{V}(\boldsymbol{R}, \boldsymbol{C}_{\mathcal{R}}, \boldsymbol{T}, \boldsymbol{C}_{\mathcal{T}}) \rangle$

$\mathcal{V}$ computes:
- $F \leftarrow_{\$} \mathbb{G}, \boldsymbol{P} \leftarrow_{\$} \mathbb{G}^{3+|\mathcal{R}|}$,
- $\boldsymbol{G}' \leftarrow_{\$} \mathbb{G}^{m-|\mathcal{R}|-3}, \boldsymbol{H} \leftarrow_{\$} \mathbb{G}^m$

$\mathcal{V} \rightarrow \mathcal{P} : F, \boldsymbol{P}, \boldsymbol{G}', \boldsymbol{H}$

$\mathcal{P}$ computes:
- $r_{A,1} \leftarrow_{\$} \mathbb{Z}_p$,
- $A_1 = F^{r_{A,1}} \cdot \hat{\boldsymbol{g}}_1^{\text{vec}(\mathbf{E})} \cdot \hat{\boldsymbol{g}}_2^{\text{vec}(\mathbf{B})} \cdot \boldsymbol{g}_3^{\boldsymbol{a}^{\mathcal{S}}} \cdot \boldsymbol{g}_4^{\boldsymbol{r}^{\mathcal{S}}} \cdot \boldsymbol{g}_5^{\boldsymbol{x}} \cdot \boldsymbol{h}_5^{\boldsymbol{x}^{\circ -1}}$

$\mathcal{P} \rightarrow \mathcal{V} : A_1$

$\mathcal{V}$ computes:
- $u, v \leftarrow_{\$} \mathbb{Z}_p$

$\mathcal{V} \rightarrow \mathcal{P} : u, v$

..... In-line Σ-protocol .....

$\mathcal{P}$ computes:
- $r_{A,2}, r_{A,3} \leftarrow_{\$} \mathbb{Z}_p, \boldsymbol{c} \leftarrow_{\$} \mathbb{Z}_p^{3+|\mathcal{R}|}$,
- $A_2 = F^{r_{A,2}} \cdot \boldsymbol{P}^{(\xi \| \eta \| 1 \| \hat{\boldsymbol{e}})}$,
- $A_3 = F^{r_{A,3}} \cdot \boldsymbol{P}^{\boldsymbol{c}}$

$\mathcal{P} \rightarrow \mathcal{V} : A_2, A_3$

$\mathcal{V}$ computes:
- $e \leftarrow_{\$} \mathbb{Z}_p$

$\mathcal{V} \rightarrow \mathcal{P} : e$

$\mathcal{P}$ computes:
- $\theta_1 = r_{A,3} + e \cdot r_{A,2}$,
- $\boldsymbol{\theta}_2 = \boldsymbol{c} + e \cdot (\xi \| \eta \| 1 \| \hat{\boldsymbol{e}})$

$\mathcal{P} \rightarrow \mathcal{V} : \theta_1, \boldsymbol{\theta}_2$

$\mathcal{V}$: Verification Procedure:
- $F^{\theta_1} \cdot \boldsymbol{P}^{\boldsymbol{\theta}_2} \stackrel{?}{=} A_3 \cdot A_2^e$

If verification passed, both prover and verifier compute and verifier compute $A = A_1 \cdot A_2 \cdot (\boldsymbol{h}_2 \| \boldsymbol{h}_3)^{-1}$.

$\mathcal{V}$ computes:
- $w \leftarrow_{\$} \mathbb{Z}_p$

$\mathcal{V} \rightarrow \mathcal{P} : w$

Continue executing the Omniring protocol after the prover receiving after the prover receiving w from the verifier as described in Section 5.1 of [22].

Protocol 3: The `outer` protocol of Omniring.

In [22], the authors deduce the desirable relations from the above equation using the Schwartz-Zipple lemma. Since the witnesses $(\boldsymbol{a}^{\mathcal{S}'}, \boldsymbol{r}^{\mathcal{S}'}, \boldsymbol{x}', \boldsymbol{E})$ are committed after send-

ing (u, v), these witnesses can be chosen according to (u, v), i.e., the coefficients are not fixed for the Schwartz-Zipple lemma. Hence, one cannot apply the Schwartz-Zipple lemma here. Meanwhile, one cannot apply the rewinding technique here due to the same reason.

We complete these steps using the soundness proof of Theorem 2. From the in-line Σ-protocol of Protocol 3, one can deduce that the witnesses$(\boldsymbol{a}^{\mathcal{S}'}, \boldsymbol{r}^{\mathcal{S}'}, \boldsymbol{x}', \boldsymbol{E})$ are committed before sending (u, v). Hence, by the rewinding technique, (25) holds for $|\mathcal{S}|$ different values of v and 3 different values of u. Then one can deduce that $\boldsymbol{R}^{\boldsymbol{e}'_l} = H^{x'_l}$, $\boldsymbol{C}_{\mathcal{R}}^{\boldsymbol{e}'_l} = G^{a_l^{\mathcal{S}'}} H^{r_l^{\mathcal{S}'}}$, and $\mathsf{tag}_l = G^{\frac{1}{x'_l}}$ for $l \in [|\mathcal{S}|]$ as desirable just like the proof of Theorem 2.

The proof size of the scheme is $2\cdot\lceil\log_2(3+|\mathcal{R}|+|\mathcal{R}||\mathcal{S}|+\beta|\mathcal{T}|+3|\mathcal{S}|)\rceil+2\lceil\log_2(4+|\mathcal{R}|)\rceil+12$ elements. The verification time is dominated by a single multi-exponentiation of size about $4|\mathcal{R}|+2|\mathcal{R}||\mathcal{S}|+2\beta|\mathcal{T}|+6|\mathcal{S}|+\lceil\log_2(3+|\mathcal{R}|+|\mathcal{R}||\mathcal{S}|+\beta|\mathcal{T}|+3|\mathcal{S}|)\rceil+\lceil\log_2(4+|\mathcal{R}|)\rceil$.

5. Application: K-out-of-N Proof of Knowledge

In this section, we construct an SHVZK protocol for K-out-of-N proof of knowledge using the general model of Section 3, where $1 \le K \le N$. More concretely, we will give an SHVZK protocol $\Pi_{K\text{-}N}$ for the following relation

$$\mathcal{R}_{K\text{-}N} = \left\{ \begin{pmatrix} g, \boldsymbol{vk}, \\ K, N \end{pmatrix} \;\middle|\; \begin{array}{c} \exists\ i_j, vk_{i_j}, sk_{i_j} \text{ for } 1 \le j \le K, \\ \text{s.t., } i_j \in [1, N] \wedge \text{ each } i_j \text{ is different} \\ \wedge\ vk_{i_j} \in \boldsymbol{vk} \wedge\ vk_{i_j} = g^{sk_{i_j}} \\ \text{for } 1 \le j \le K \end{array} \right\} \tag{26}$$

where $\boldsymbol{vk} = (vk_1, \cdots, vk_N) \in \mathbb{G}^N$ and $g \in \mathbb{G}$.

One can deduce $\mathcal{R}_{K\text{-}N}$ from $\mathcal{R}_{\texttt{KKDL}}$ of (1) as follows.

1) In $\mathcal{R}_{\texttt{KKDL}}$, let $m = K$, $n_1 = N$, $n_2 = 1$, $k = 1$, $f(\boldsymbol{b}_L) = \boldsymbol{b}_L \circ \boldsymbol{b}_L - \boldsymbol{b}_L = \boldsymbol{0}^N$, $\phi_1(\boldsymbol{b}_L) = \sum_{j=1}^{N} b_{L,j}$, $d_1 = K$, $\boldsymbol{q}_2 = \{g\}$ and $\boldsymbol{\psi}_i = \{-sk_i\}$ for $1 \le i \le N$. Meanwhile, let $\boldsymbol{q}_{1,i} = \{0, \cdots, 0, vk_i, 0, \cdots, 0\}$ for $1 \le i \le N$, where the i^{th} component of $\boldsymbol{q}_{1,i}$ is vk_i and the other components are zero.
2) Since $f(\boldsymbol{b}_L) = \boldsymbol{b}_L \circ \boldsymbol{b}_L - \boldsymbol{b}_L = \boldsymbol{b}_L \circ (\boldsymbol{b}_L - \boldsymbol{1}^N) = \boldsymbol{0}^N$, each component of $\boldsymbol{b}_L$ is either 0 or 1. Since $\phi_1(\boldsymbol{b}_L) = \sum_{j=1}^{N} b_{L,j} = K$, $\boldsymbol{b}_L$ has exactly K components which are "1". Denote $i_1, \cdots, i_K$ as the positions where the components of b_L are "1", and denote $\boldsymbol{S} = \{i_1, \cdots, i_K\}$.
3) By the analysis above, one has that

$$\boldsymbol{q}_{1,j}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_j} = \begin{cases} vk_j \cdot g^{-sk_j} = 1 & j \in \boldsymbol{S} \\ vk_j^0 \cdot g^{-sk_j \cdot 0} = 1 & j \notin \boldsymbol{S} \end{cases} \tag{27}$$

The case $j \in \boldsymbol{S}$ above means that the prover knows the DL relation between vk_j and the generator g, and the case $j \notin \boldsymbol{S}$ is trivial that the prover does not need to consider in the protocol. Hence, $\mathcal{R}_{K\text{-}N}$ can be directly deduced.

Then, one can use $\Pi_{\texttt{KKDL}}$ and $\Pi_{\texttt{inner}}$ to construct a zero-knowledge protocol $\Pi_{K\text{-}N}$ for $\mathcal{R}_{K\text{-}N}$. The security of $\Pi_{K\text{-}N}$ is shown in Theorem 3.

The total proof size of $\Pi_{K\text{-}N}$ is $2 \cdot \lceil\log_2(N)\rceil + 14$ using the improved inner-product argument and the Fiat-Shamir heuristic (since n_2 and $|\boldsymbol{\eta}| = 1$, we do not use the improved inner-product argument to θ_2 of Protocol 2 and $\boldsymbol{\eta}$ of Protocol 1). The verifier's cost is dominated by a single multi-exponentiation of size $2N + \lceil\log_2(N)\rceil$.

Theorem 3 (Security of $\Pi_{K\text{-}N}$)**.** *The protocol $\Pi_{K\text{-}N}$ has perfect completeness, perfect special honest-verifier zero-knowledge, and computational witness-extended emulation.*

The proof of Theorem 3 can be deduced from the proofs of Theorem 2 and Theorem 1.

Remark. *In the case $K = 1$, it is a one-out-of-N proof which implies ring signatures. The details can be found in Appendix D.*

References

[1] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, ser. STOC '85. New York, NY, USA: Association for Computing Machinery, 1985, p. 291–304.

[2] G. Maxwell, "Confidential transactions. https://people.xiph.org/~greg/confidential_values.txt."

[3] S. Noether, A. Mackenzie *et al.*, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.

[4] G. Fuchsbauer, M. Orrù, and Y. Seurin, "Aggregate cash systems: A cryptographic investigation of mimblewimble," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 657–689.

[5] E. Ben Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, pp. 459–474.

[6] B. Parno, C. Gentry, J. Howell, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," Cryptology ePrint Archive, Report 2013/279, 2013, https://ia.cr/2013/279.

[7] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, "Snarks for c: Verifying program executions succinctly and in zero knowledge," Cryptology ePrint Archive, Report 2013/507, 2013, https://ia.cr/2013/507.

[8] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," Cryptology ePrint Archive, Report 2013/879, 2013, https://ia.cr/2013/879.

[9] J. Bootle, A. Cerulli, J. Groth, S. Jakobsen, and M. Maller, "Nearly linear-time zero-knowledge proofs for correct program execution," Cryptology ePrint Archive, Report 2018/380, 2018, https://ia.cr/2018/380.

[10] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham, "Delegatable anonymous credentials," Cryptology ePrint Archive, Report 2008/428, 2008, https://ia.cr/2008/428.

[11] J. Camenisch and T. Groß, "Efficient attributes for anonymous credentials (extended version)," Cryptology ePrint Archive, Report 2010/496, 2010, https://ia.cr/2010/496.

[12] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," Cryptology ePrint Archive, Report 2013/622, 2013, https://ia.cr/2013/622.

[13] J. Camenisch, M. Dubovitskaya, K. Haralambiev, and M. Kohlweiss, "Composable and modular anonymous credentials: Definitions and practical constructions," Cryptology ePrint Archive, Report 2015/580, 2015, https://ia.cr/2015/580.

[14] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," Cryptology ePrint Archive, Report 2017/1066, 2017, https://eprint.iacr.org/2017/1066.

[15] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 315–334.

[16] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit, "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting," Cryptology ePrint Archive, Report 2016/263, 2016, https://ia.cr/2016/263.

[17] T. Attema and R. Cramer, "Compressed sigma-protocol theory and practical application to plug & play secure algorithmics," in *Advances in Cryptology – CRYPTO 2020*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 513–543.

[18] T. Attema, R. Cramer, and S. Fehr, "Compressing proofs of k-out-of-n partial knowledge," in *Advances in Cryptology – CRYPTO 2021*. Springer International Publishing, 2021, pp. 65–91.

[19] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology — CRYPTO '91*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 129–140.

[20] F. Bao, R. H. Deng, and H. Zhu, "Variations of diffie-hellman problem," in *International conference on information and communications security*. Springer, 2003, pp. 301–312.

[21] T. Icart, "How to hash into elliptic curves," Cryptology ePrint Archive, Paper 2009/226, 2009, https://eprint.iacr.org/2009/226. [Online]. Available: https://eprint.iacr.org/2009/226

[22] R. W. Lai, V. Ronge, T. Ruffing, D. Schröder, S. A. K. Thyagarajan, and J. Wang, "Omniring: Scaling private payments without trusted setup," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19, 2019, pp. 31–48.

[23] T. H. Yuen, S.-F. Sun, J. K. Liu, M. H. Au, M. F. Esgin, Q. Zhang, and D. Gu, "Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security," in *Financial Cryptography and Data Security*, 2020, pp. 464–483.

[24] S. Noether, "Ring signature confidential transactions for monero," Cryptology ePrint Archive, Report 2015/1098, 2015, https://ia.cr/2015/1098.

[25] T. H. Yuen, S. feng Sun, J. K. Liu, M. H. Au, M. F. Esgin, Q. Zhang, and D. Gu, "Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security," Cryptology ePrint Archive, Report 2019/508, 2019, https://ia.cr/2019/508.

[26] J. Groth and Y. Ishai, "Efficient zero-knowledge argument for correctness of a shuffle," in *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, 2008.

[27] E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in *Advances in Cryptology — CRYPTO 2002*. Springer Berlin Heidelberg, 2002.

[28] B. E. Diamond, "Many-out-of-many proofs and applications to anonymous zether," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1800–1817.

[29] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh, "Zether: Towards privacy in a smart contract world," in *Financial Cryptography and Data Security*, 2020, pp. 423–443.

[30] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International conference on the theory and application of cryptology and information security*. Springer, 2001, pp. 552–565.

[31] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit, "Short accountable ring signatures based on ddh," in *Computer Security – ESORICS 2015*. Springer International Publishing, 2015, pp. 243–265.

[32] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp. 62–73.

[33] S. Turner, D. Brown, K. Yiu, R. Housley, and T. Polk, "Rfc5480: Elliptic curve cryptography subject public key information. https://www.rfc-editor.org/rfc/rfc5480.txt."

[34] Y. Guo, H. Karthikeyan, A. Polychroniadou, and C. Huussin, "Pride ct: Towards public consensus, private transactions, and forward secrecy in decentralized payments," Cryptology ePrint Archive, Paper 2023/1948, 2023, https://eprint.iacr.org/2023/1948. [Online]. Available: https://eprint.iacr.org/2023/1948

[35] Wikipedia contributors, "Tor (network) — Wikipedia, the free encyclopedia," 2024, [Online; accessed 23-March-2024]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Tor_(network)&oldid=1213947809

[36] J. Rising, "What are paymasters." [Online]. Available: https://www.stackup.sh/blog/what-are-paymasters

[37] Y. Jia, S.-F. Sun, Y. Zhang, Q. Zhang, N. Ding, Z. Liu, J. K. Liu, and D. Gu, "pbtpbt: A new privacy-preserving payment protocol for blockchain transactions," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 647–662, 2022.

[38] J. Groth and M. Kohlweiss, "One-out-of-many proofs: Or how to leak a secret and spend a coin," in *Advances in Cryptology - EUROCRYPT 2015*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 253–280.

[39] M. Chase and A. Lysyanskaya, "On signatures of knowledge," in *Annual International Cryptology Conference*. Springer, 2006, pp. 78–96.

[40] T. Zheng, S. Gao, B. Xiao, and Y. Song, "Leaking arbitrarily many secrets: Any-out-of-many proofs and applications to ringct protocols," *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2533–2550, 2023.

[41] Wikipedia contributors, "Schwartz–zippel lemma — Wikipedia, the free encyclopedia," 2022, [Online; accessed 25-March-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Schwartz%E2%80%93Zippel_lemma&oldid=1065325812

[42] I. Damgård, "On Sigma-protocol. https://www.cs.au.dk/~ivan/Sigma.pdf."

Appendix A. Application: Multi-Receiver Anonymous Zether

In Anonymous Zether, a sender may hide herself and the receivers in a larger ring $\vec{\boldsymbol{R}} = \{pk_i, 0 \leq i \leq N-1\}$. The original paper [28] only consider the case that the number of receiver is one, but it may be larger than one. In this paper, we consider the multi-receiver scenario. Although [34] also considers the multi-receiver scenario, they do not consider the anonymity of sender since an artifact of Ethereum where invocation to smart contract trivially reveals the identity of the invoking party. However, the identity of the invoking party can be different from the identity of the smart contract, and we can trivially use Tor network [35], paymaster [36] and ring signature to avoid this.

To an observer, it should be impossible to discern which among a ring's members sent or received funds. Specifically, a sender should choose indices l_s and $(l_{r,0}, \cdots, l_{r,\iota})$ for which pk_{l_s} and $(pk_{l_{r,0}}, \cdots, pk_{l_{r,\iota}})$ belong to the sender and ι receivers, respectively. Let $\vec{a} = \{a_0, \cdots, a_{N-1}\}$ be a list of amounts where a_i is the amount for the user with pk_i. Indeed, we don't need to consider the number of receivers, we only need to consider that the amounts is non-negative except the amount of sender, and the sum of amounts for all uses is zero, i.e., $\sum_{i=0}^{N-1} a_i = 0$ with $a_{l_s} = -a \leq 0$ and $a_j \geq 0$ for $j \neq l_s$. Let $\vec{\boldsymbol{C}} = \{C_0, \cdots, C_{N-1}\}$ and D be the list of ciphertexts that encrypt $\vec{a}$ using $\vec{\mathbf{R}}$.

To apply the transfer, the contract should homomorphically add (C_i, D) to pk_i's balance for each i; we denote the list of new balances by $(C_{L,i}, C_{R,i})_{i=0}^{N-1}$. Denote $\vec{\boldsymbol{C}}_L$ and $\vec{\boldsymbol{C}}_R$ be the vector of $C_{L,i}$ and $C_{R,i}$, respectively.

The relations we need to prove are:

- The indexes of sender and receivers are in the proper ranges, i.e., $l_s \in [0, N-1]$.
- Sender knows the secret key, i.e., $\exists sk$, s.t., $pk_{l_s} = g^{sk}$.
- $\exists r$, s.t., sender knows the randomness, i.e., $D = g^r$;
- The sum of all amounts are zero, i.e., $\sum_{i=0}^{N-1} a_i = 0$.
- The amounts of receivers and the account balance of sender are in the proper range, i.e., $a_i \in \{0, \cdots, \text{MAX}\}$ for $i \neq l_s$ and $a_s \in \{0, \cdots, \text{MAX}\}$ where a_s is the remaining account balance of sender.
- The epoch of this transaction is correct.

In conclusion, we will give a SHVZK protocol Π_{AZ} for the following relation

$$\mathcal{R}_{AZ} = \left\{ (\vec{\boldsymbol{R}}, \vec{\boldsymbol{C}}, \vec{\boldsymbol{C}}_L, \vec{\boldsymbol{C}}_R, D, C_{ep}, g_{ep}) \;\middle|\; \begin{array}{c} \exists\ l_s, sk, r, a_s, \vec{a} \\ \text{s.t., } l_s \in [0, N-1] \wedge \\ C_i = g^{a_i} pk_i^r \text{ for } \forall i \in [0, N-1] \\ \wedge\ D = g^r \wedge\ pk_{l_s} = g^{sk} \wedge \\ C_{L,l_s} = g^{a_s} C_{R,l_s}^{sk} \wedge\ C_{ep} = g_{ep}^{sk} \\ \wedge \sum_{i=0}^{N-1} a_i = 0 \wedge\ a_s, a_j \in [0, \text{MAX}] \\ \text{for } \forall j \in [0, N-1] \end{array} \right\} \tag{28}$$

To prove (28), one can construct one unit vector $\vec{e}_0$ of length N where the l_s-th position of $\vec{e}_0$ is 1 and 0 otherwise. For $\vec{a}' = \{a'_0, \cdots, a'_{N-1}\}$, construct an auxiliary vector $\vec{a}'$ as follows:

$$\begin{cases} a'_{l_s} = -a_{l_s} \\ a'_i = a_i, \text{otherwise} \end{cases} \tag{29}$$

Meanwhile, construct N unit vectors of length β where $\vec{b}_i$ is the binary representation of a'_i, and $\vec{b}_s$ is the unit vector of length β where $\vec{b}_s$ is the binary representation of a_s.

Construct the witness vectors $\vec{c}_L$ and $\vec{c}_R$ as equation (30) where $\xi = \langle \vec{v}^N, \vec{a} \rangle + u \cdot a_s + u^2 \cdot sk + u^3 \cdot r$, $\vec{sk}$ is a vector of N reputations of sk, $\vec{r}$ is a vector of N reputations of r, and $\vec{\boldsymbol{E}} = (\vec{b}_0 || \vec{b}_1 || \cdots || \vec{b}_{N-1})$.

Hence, (28) can be transformed into inner equations and outer equations as follows:

- For the outer equations:
  1) The amounts are correct: $\vec{\boldsymbol{C}}^{\vec{v}^N} = g^{\langle \vec{v}^N, \vec{a} \rangle} \vec{\boldsymbol{R}}^{\vec{v}^n \circ \vec{r}}$.
  2) The account balance of sender is correct: $\vec{\boldsymbol{C}}_L^{\vec{e}_0} = g^{a_s} \vec{\boldsymbol{C}}_R^{sk \cdot \vec{e}_0}$.
  3) Sender knows the private key: $\vec{\boldsymbol{R}}^{\vec{e}_0} = g^{sk}$.
  4) The construction of D is correct: $D = g^r$.
  5) Epoch is correct: $C_{ep} = g_{ep}^{sk}$.
- For the inner equations, construct the vectors as (32):
  1) The components of binary representations are zero or one, i.e., $\vec{e}_0$, $\vec{b}_i$ $(0 \leq i \leq N-1)$ and $\vec{b}_s$ are unit vectors: $\langle \vec{c}_L, \vec{c}_R \circ \vec{v}_0 \rangle = 0$ and $\langle \vec{c}_L - \vec{c}_R - \vec{1}^{N+(N+1)\beta}, \vec{u}_{17} \rangle = 0$.
  2) $\vec{e}_0$ is unit vectors with only one position which is one: $\langle \vec{c}_L, \vec{u}_4 \rangle = 1$.
  3) Construct an auxiliary vector $\vec{a}'$ satisfying (29). This condition can be divided into the following two conditions:
     - The relation between the seventh, tenth and eleventh positions of $\vec{c}_R$ are correct: $\langle \vec{c}_R, \vec{w}_{13} \rangle = \langle \vec{1}^N, \vec{y}^N \rangle$ and $\langle \vec{c}_R, \vec{w}_{14} \rangle = 0$.
     - $a_{l_s} = -a'_{l_s}$: $\langle \vec{c}_L, \vec{c}_R \circ \vec{v}_2 \rangle = 0$.
     - $a_i = a'_i$ for $i \neq l_s$: $\langle \vec{c}_L, \vec{c}_R \circ \vec{v}_3 \rangle + \langle \vec{c}_L, \vec{u}_3 \rangle = 0$.
  4) $\vec{b}_i$ and $\vec{b}_s$ is the binary representation of a'_i and a_s for $i \in [0, N-1]$: $\langle \vec{c}_L, \vec{u}_5 \rangle = 0$.
  5) The sum of all amounts is zero, i.e., $\sum_{i=0}^{N-1} a_i = 0$: $\langle \vec{c}_L, \vec{u}_6 \rangle = 0$.
  6) Since the outer equations are fulfilled using the relation between $\vec{c}_L$ and $\vec{\boldsymbol{G}}_w$ of (31), the following conditions ensure the correctness of the construction of $\vec{c}_L$:
     - The first position of $\vec{c}_L$ is correct, i.e., $\xi = \langle \vec{v}^N, \vec{a} \rangle + u \cdot a_s + u^2 \cdot sk + u^3 \cdot r$: $\langle \vec{c}_L, \vec{u}_7 \rangle = 0$.
     - The second position of $\vec{c}_L$ is $\vec{r}^N$: $\langle \vec{c}_L, \vec{u}_8 \rangle = 0$.
     - The third position of $\vec{c}_L$ is $\vec{1}^N$: $\langle \vec{c}_L, \vec{u}_9 \rangle = \langle \vec{1}, \vec{y}^N \rangle$.
     - The fourth and seventh positions of $\vec{c}_L$ are equal: $\langle \vec{c}_L, \vec{u}_{10} \rangle = 0$.
     - The fifth position of $\vec{c}_L$ is $sk \cdot \vec{e}_0$, this condition can be fulfilled by the following two conditions:
       * The fourth position of $\vec{c}_R$ consists of N reputations of the last position of $\vec{c}_L$, i.e., The fourth position of $\vec{c}_R$ is $\vec{sk}^N$: $\langle \vec{c}_L, \vec{u}_{12} \rangle + \langle \vec{c}_R, \vec{w}_{12} \rangle = 0$.
       * The fifth positions of $\vec{c}_R$ are $\vec{1}^N$: $\langle \vec{b_R}, \vec{w}_{15} \rangle =$

$$\begin{aligned}\vec{c}_L &= (\xi, \quad \vec{r}^N,\vec{1}^N, \quad \vec{e}_0,sk\cdot\vec{e}_0, \quad sk,\vec{e}_0, \quad \vec{\boldsymbol{E}},\vec{b}_s, \quad \vec{a},\vec{a}', \quad a_s,r, \quad sk)\\ \vec{c}_R &= (0, \quad \vec{0}^N,\vec{0}^N, \quad \vec{sk}^N,\vec{1}^N, \quad 0,\vec{e}_0-\vec{1}^N, \quad \vec{\boldsymbol{E}}-\vec{1}^{N\beta},\vec{b}_s-\vec{1}^{\beta}, \quad \vec{e}_0,\vec{e}_0, \quad 0,0, \quad 0)\end{aligned} \tag{30}$$

$\langle\vec{1}^N,\vec{y}^N\rangle$.

$*$ $sk\cdot\vec{e}_0\circ\vec{1}^N=\vec{e}_0\circ\vec{sk}$: $\langle\vec{c}_L,\vec{c}_R\circ\vec{v}_1\rangle=0$.

– The sixth position of $\vec{c}_L$ is sk, i.e., it equals the last position: $\langle\vec{c}_L,\vec{u}_{11}\rangle=0$.

Combing all the six condition of the outer equations using the powers of randomness u, one can deduce the generator

$$\begin{aligned}\vec{\boldsymbol{G}}_w = ((g||\vec{\boldsymbol{R}}^{\vec{v}^n}||\vec{\boldsymbol{C}}^{-\vec{v}^n}\cdot D^{u^3}\cdot C_{epoch}^{u^4}||\\ \vec{\boldsymbol{C}}_L^{\circ u}\circ\vec{\boldsymbol{R}}^{\circ u^2}||\vec{\boldsymbol{C}}_R^{\circ u}||g_{epoch}^{u^4})^{\circ w}\circ\vec{\boldsymbol{P}}||\vec{\boldsymbol{G}}').\end{aligned} \tag{31}$$

The dimension of $\vec{\boldsymbol{G}}_w$, $\vec{\boldsymbol{G}}'$ and $\vec{\boldsymbol{P}}$ is $m=7N+(N+1)\beta+5$, $3N+(N+1)\beta+3$ and $4N+2$, respectively.

A.1. The Protocol

Let $\vec{\theta}=\sum_{i=0}^{3}z^i\cdot\vec{v}_i$, $\vec{\zeta}=\sum_{i=3}^{12}z^i\cdot\vec{u}_i$, $\vec{\mu}=\sum_{i=3}^{12}z^i\cdot\vec{u}_i+z^{16}\cdot\vec{u}_{16}$, $\vec{\nu}=z^{16}\cdot\vec{u}_{16}$, $\vec{\omega}=\sum_{i=12}^{15}z^i\cdot\vec{w}_i$, $\vec{\alpha}=\vec{\theta}^{-1}\circ(\vec{\omega}-\vec{\nu})$, $\vec{\beta}=\vec{\theta}^{-1}\circ\vec{\mu}$, $l(X)=\vec{c}_L+\vec{\alpha}+\vec{s}_L\cdot X$, $\vec{r}(X)=\vec{\theta}\circ(\vec{c}_R+\vec{s}_R\cdot X)+\vec{\mu}$ and $t(X)=t_2X^2+t_1X+t_0$, one can deduce equation (33).

Let $\vec{\boldsymbol{G}}'=(\vec{g}_0,\vec{g}_1,\vec{g}_2,\vec{g}_3,\vec{g}_4,g_5,g_6,g_7)\leftarrow_{\$}\mathbb{G}^{3N+(N+1)\beta+3}$ where $\vec{g}_0,\vec{g}_3,\vec{g}_4\in\mathbb{G}^N$, $\vec{g}_1\in\mathbb{G}^{N\beta}$, and $\vec{g}_2\in\mathbb{G}^{\beta}$. Also, let $\vec{\boldsymbol{H}}=(\vec{h}_0,\vec{h}_1,\vec{h}_2,\vec{h}_3,\vec{h}_4,\vec{h}_5,\vec{h}_6)\leftarrow_{\$}\mathbb{G}^m$ where $\vec{h}_0\in\mathbb{G}^{4N+2}$, $\vec{h}_1\in\mathbb{G}^N$, $\vec{h}_2\in\mathbb{G}^{N\beta}$, $\vec{h}_3\in\mathbb{G}^{\beta}$, $\vec{h}_4\in\mathbb{G}^N$, $\vec{h}_5\in\mathbb{G}^N$, and $\vec{h}_6\in\mathbb{G}^3$. Denote $\vec{g}_0'=\vec{g}_0\circ\vec{h}_1\circ\vec{h}_3\circ\vec{h}_4$, $\vec{g}_1'=\vec{g}_1\circ\vec{h}_2$, and $\vec{g}_2'=\vec{g}_2\circ\vec{h}_3$. The detailed protocol is shown in Protocol 4.

Appendix B. Cryptanalysis of RingCT-3.0

In FC 2020, Yuen et al. proposed RingCT-3.0 [23], [25] for blockchain confidential transactions to replace the RingCT-1.0 of Monero [24] with a shorter size and stronger security.

$$\begin{aligned}&A_1\cdot A_2\cdot(\boldsymbol{h}_2\|\boldsymbol{h}_3)^{-1}\\ =&F^{r_A}\cdot\hat{\boldsymbol{g}}_1^{\mathrm{vec}(\mathbf{E})}\cdot\hat{\boldsymbol{g}}_2^{\mathrm{vec}(\mathbf{B})}\cdot\boldsymbol{g}_3^{\boldsymbol{a}^{\mathcal{S}}}\cdot\boldsymbol{g}_4^{\boldsymbol{r}^{\mathcal{S}}}\cdot\boldsymbol{g}_5^{\boldsymbol{x}}\cdot\\ &\boldsymbol{h}_5^{\boldsymbol{x}^{\circ-1}}\cdot(\boldsymbol{h}_2\|\boldsymbol{h}_3)^{-1}\cdot\boldsymbol{P}^{(\xi\|\eta\|1\|\hat{\mathbf{e}})}\\ =&F^{r_A}\cdot(\boldsymbol{P}||\boldsymbol{G}')^{\boldsymbol{c}_L}\cdot\boldsymbol{H}^{\boldsymbol{c}_R}\\ =&A.\end{aligned} \tag{34}$$

B.1. Basic Construction of RingCT-3.0

In [25], the authors consider the case of multiple input coins and output coins. For ease of explanation, we consider the case of one input coin and one output coin, i.e., $k=1$ in Section 5.1 of [25]. One can easily extend our cryptanalysis to the case of multiple coins.

Let N denote the size of the ring, $\boldsymbol{vk}=(vk_1,\cdots,vk_N)$ be the verification keys of input coins (the ring), and $\boldsymbol{c}=(c_1,\cdots,c_N)$ be the corresponding commitments of the amounts of input coins. The relation we need to prove for RingCT-3.0 is

$$\mathcal{R}_{\text{RCT-3.0}}=\left\{\begin{array}{c|c}(\boldsymbol{c},\tilde{c},\boldsymbol{vk},u,\\ g_c,h_c,U)&\begin{array}{c}\exists\ \mathtt{ind},\tilde{v},\tilde{r},\Delta,sk,\text{ s.t.,}\\ \tilde{c}=g_c^{\tilde{r}}h_c^{\tilde{v}}\ \wedge\ \tilde{v}\in[0,V_{\max}]\\ \wedge\ vk_{\mathtt{ind}}=g^{sk}\wedge\ U=u^{\frac{1}{sk}}\\ \wedge\ c_{\mathtt{ind}}/\tilde{c}=g_c^{\Delta}\end{array}\end{array}\right\}, \tag{35}$$

where $\mathtt{ind}\in[1,N]$ is the index of the ring that the prover knows the secret key sk of $vk_{\mathtt{ind}}$ and the opening of the commitment $c_{\mathtt{ind}}$, $\tilde{c}\in\mathbb{G}$ is the commitment of the output amount $\tilde{v}\in\mathbb{Z}_p$ using randomness $\tilde{r}\in\mathbb{Z}_p$, $V_{\max}$ is the maximum number of the amount, and $U\in\mathbb{G}$ is the key image. One can split relation (35) into the following three parts.

1) $\tilde{c}=g_c^{\tilde{r}}h_c^{\tilde{v}}$ and $\tilde{v}\in[0,V_{\max}]$ mean that the commitment of the output coin is well-formed, and the amount is in the proper range. This relation can be fulfilled using Bulletproofs, so we do not consider it here.
2) $vk_{\mathtt{ind}}=g^{sk}$ means that the prover knows the private key sk corresponding to one verification key in the ring. $U=u^{\frac{1}{sk}}$ means that the key image is well-formed.
3) $c_{\mathtt{ind}}/\tilde{c}=g_c^{\Delta}$ means that the input amount equals the output amount.

Let $\hat{\boldsymbol{g}},\boldsymbol{h}\in\mathbb{G}^N$ and $g,h,g_c,u,h_c\in\mathbb{G}$ be the generators with unknown DL relations. Let $\boldsymbol{b}_L\in\mathbb{Z}_p^N$ be a binary vector where $b_{L,j}=1$ when $j=\mathtt{ind}$ and $b_{L,j}=0$ otherwise. The zero-knowledge protocol of RingCT-3.0 without the range proof part is shown in Protocol 5 where $\delta(y,z,w)=z^2+w(z-z^2)\cdot\langle\mathbf{1}^N,\boldsymbol{y}^N\rangle-z^3\cdot\langle\mathbf{1}^N,\mathbf{1}^N\rangle$.

B.2. Cryptanalysis of RingCT-3.0.

Next, we will construct an attack on RingCT-3.0, which can pass the verification procedure without knowing any secret key of $\boldsymbol{vk}$ and any opening of $\boldsymbol{c}$.

1) Choose an amount $v'\in[0,V_{\max}]$ and two random numbers $\tilde{r}',r'\in\mathbb{Z}_p$, construct a commitment $\tilde{c}'=g_c^{\tilde{r}'}h_c^{v'}$ for the new output coin, and a commitment $c'=g_c^{r'}h_c^{v'}$ for a "forged" input coin (a coin never existed in the chain). Since $v'\in[0,V_{\max}]$, one can construct a correct range proof for $\tilde{c}'$. Denote $\Delta'=r'-\tilde{r}'$.
2) Choose a random secret key $sk'\in\mathbb{Z}_p$, and generate a verification key $vk'=g^{sk'}$ for the "forged" input coin. Meanwhile, generate the key image $U'=u^{\frac{1}{sk'}}$. Choose an input coin from the ring, and denote its index as $\mathtt{ind}$ (without knowing its secret key, amount, and randomness). Construct $\boldsymbol{b}_L$ and $\boldsymbol{b}_R$ as RingCT-3.0 using $\mathtt{ind}$.

$$
\begin{aligned}
\vec{v}_0 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\vec{y}^{N+(N+1)\beta}, \quad , \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{v}_1 &= (\cdot, \quad \cdot,\cdot, \quad \vec{y}^N,-\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{v}_2 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \vec{y}^N,\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{v}_3 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \vec{y}^N,-\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_3 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad -\vec{y}^N,\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_4 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\vec{1}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_5 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \vec{y}^{N+1}\otimes\vec{2}^\beta, \quad \cdot,-\vec{y}^{N+1}, \quad ,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_6 &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \vec{1}^N,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_7 &= (1, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad -\vec{v}^N,\cdot, \quad -u,-u^3, \quad -u^2)\\
\vec{u}_8 &= (\cdot, \quad \vec{y}^N,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,-\sum_{i=0}^{N-1} y^i, \quad \cdot)\\
\vec{u}_9 &= (\cdot, \quad \cdot,\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_{10} &= (\cdot, \quad \cdot,\cdot, \quad \vec{y}^N,\cdot, \quad \cdot,-\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_{11} &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad y,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad -y)\\
\vec{u}_{12} &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad -\sum_{i=0}^{N-1} y^i)\\
\vec{w}_{12} &= (\cdot, \quad \cdot,\cdot, \quad \vec{y}^N,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{w}_{13} &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,-\vec{y}^N, \quad \cdot,\cdot, \quad \vec{y}^N,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{w}_{14} &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \vec{y}^N,-\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{w}_{15} &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\vec{y}^N, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)\\
\vec{u}_{16} &= (\cdot, \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot,\vec{y}^{N+(N+1)\beta}, \quad , \quad \cdot,\cdot, \quad \cdot,\cdot, \quad \cdot)
\end{aligned}
\tag{32}
$$

$$
\begin{aligned}
&t_0 = \langle \vec{c}_L, \vec{\theta}\circ\vec{c}_R\rangle + \langle \vec{\alpha}, \vec{\theta}\circ\vec{c}_R\rangle + \langle \vec{c}_L, \vec{\mu}\rangle + \langle \vec{\alpha}, \vec{\mu}\rangle\\
=&\sum_{i=0}^{3} z^i\cdot\langle \vec{c}_L, \vec{c}_R\circ\vec{v}_i\rangle + \sum_{i=12}^{15} z^i\cdot\langle \vec{c}_R, \vec{w}_i\rangle - z^{16}\cdot\langle \vec{c}_R, \vec{u}_{16}\rangle + \sum_{i=3}^{12} z^i\cdot\langle \vec{c}_L, \vec{u}_i\rangle + z^{16}\cdot\langle \vec{c}_L, \vec{u}_{16}\rangle + \langle \vec{\alpha}, \vec{\mu}\rangle\\
=&z^4 + z^9\cdot\langle \vec{1}^N, \vec{y}^N\rangle + z^{13}\cdot\langle \vec{1}^N, \vec{y}^N\rangle + z^{15}\cdot\langle \vec{1}^N, \vec{y}^N\rangle + z^{16}\cdot\langle \vec{1}^{N+(N+1)\beta}, \vec{y}^{N+(N+1)\beta}\rangle\\
=&\delta
\end{aligned}
\tag{33}
$$

3) We give the protocol for the "forged" proof in Protocol 5. The goal of the "forged" proof is that the attacker can legally spend a coin that never existed. The main drawback of RingCT-3.0 is that the verifier cannot ensure B_2 consists of $(h, \hat{g}_{\mathtt{ind}})$ or $(h, \hat{g}_{\mathtt{ind}}, vk_{\mathtt{ind}}, vk', c_{\mathtt{ind}}, c')$ by the hiding property of the commitment scheme. Meanwhile, the construction of B_2 cannot be fixed by the verification procedure and defaults to be h and $\hat{g}_{\mathtt{ind}}$ in the soundness proof of RingCT-3.0 (Appendix A of [25]). The main idea of our cryptanalysis is that one can replace the commitment and the verification key of a legal input coin in the ring with a "forged" commitment and a "forged" verification key silently as follows.
   a) After receiving d_1, d_2 from the verifier, the prover computes $B_2' = h^{\alpha_2}\cdot\hat{g}_{\mathtt{ind}}\cdot(vk_{\mathtt{ind}}\cdot vk'^{-1})^{\frac{1}{d_2}}\cdot(c_{\mathtt{ind}}\cdot c'^{-1})^{\frac{d_1}{d_2}}$ instead of $B_2 = h^{\alpha_2}\cdot\hat{g}_{\mathtt{ind}}$. Meanwhile, the prover computes $z'_{sk} = r_{sk} + sk'\cdot x$ and $z'_\Delta = r_\Delta + \Delta'\cdot x$ instead of $z_{sk} = r_{sk}+sk\cdot x$ and $z_\Delta = r_\Delta+\Delta\cdot x$. During the verification procedure, since
$$
\begin{aligned}
&S_1\cdot(B_1\cdot(\tilde{c}')^{-d_1}\cdot B_2'^{-d_2})^x\\
=&h^{z_{\alpha_1}-d_2z_{\alpha_2}}\cdot g^{r_{sk}}\cdot g_c^{d_1r_\Delta}\cdot[vk_{\mathtt{ind}}\cdot c_{\mathtt{ind}}^{d_1}\cdot g_{\mathtt{ind}}^{d_2}\cdot(\tilde{c}')^{-d_1}\\
&\cdot g_{\mathtt{ind}}^{-d_2}\cdot(vk_{\mathtt{ind}}^{-1}\cdot vk')\cdot(c_{\mathtt{ind}}^{-d_1}\cdot c'^{d_1})]^x\\
=&h^{z_{\alpha_1}-d_2z_{\alpha_2}}\cdot g^{r_{sk}}\cdot g_c^{d_1r_\Delta}\cdot vk'^x\cdot(c'/\tilde{c}')^{xd_1}\\
=&h^{z_{\alpha_1}-d_2z_{\alpha_2}}g^{z'_{sk}}g_c^{d_1z'_\Delta},
\end{aligned}
$$
   the prover can eliminate $vk_{\mathtt{ind}}$ and $c_{\mathtt{ind}}$ from B_1 using B_2', and append the verification key vk' and the commitment c' of the "forged" coin into B_1. Meanwhile, since sk', Δ' and the opening of $\tilde{c}'$ are known, the proof can pass the fifth equality of the verification procedure.
   b) Since $S_3'\cdot u^x = U'^{r_{sk}+x\cdot sk'} = U'^{z'_{sk}}$, the proof can pass the sixth equality of the verification procedure.

$\Pi_{\mathbb{AZ}} : \langle \mathcal{P}(\vec{\boldsymbol{R}}, \vec{\boldsymbol{C}}, \vec{\boldsymbol{C}}_L, \vec{\boldsymbol{C}}_R, D, C_{ep}, g_{ep}; l_s, sk, r, a_s, \vec{\boldsymbol{a}}),$
$\mathcal{V}(\vec{\boldsymbol{R}}, \vec{\boldsymbol{C}}, \vec{\boldsymbol{C}}_L, \vec{\boldsymbol{C}}_R, D, C_{ep}, g_{ep})\rangle$

$\mathcal{V}$ computes:
$F \leftarrow\$ \mathbb{G}, \vec{\boldsymbol{P}} \leftarrow\$ \mathbb{G}^{5N+2}, \quad \vec{\boldsymbol{G}}' \leftarrow\$ \mathbb{G}^{3N+2\beta+4}, \vec{\boldsymbol{H}} \leftarrow\$ \mathbb{G}^m$
$\mathcal{V} \rightarrow \mathcal{P} : F, \vec{\boldsymbol{P}}, \vec{\boldsymbol{G}}', \vec{\boldsymbol{H}}$
$\mathcal{P}$ computes:
$r_{A,1} \leftarrow\$ \mathbb{Z}_p, A_1 = F^{r_{A,1}} \cdot \vec{\boldsymbol{g}}'^{\vec{\boldsymbol{e}}_0} \cdot \vec{\boldsymbol{g}}'^{\vec{\boldsymbol{E}}} \cdot \vec{\boldsymbol{g}}_2'^{\vec{\boldsymbol{b}}_s} \cdot \vec{\boldsymbol{g}}_3^{\vec{\boldsymbol{a}}} \cdot \vec{\boldsymbol{g}}_4^{\vec{\boldsymbol{a}}'} g_5^{a_s} \cdot g_6^{r} \cdot g_7^{sk}$
$\mathcal{P} \rightarrow \mathcal{V} : A_1$
$\mathcal{V}$ computes:
$u, v \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : u, v$

.....In-line Σ-protocol.....

$\mathcal{P}$ computes:
$r_{A,2}, r_{A,3} \leftarrow\$ \mathbb{Z}_p, \vec{\boldsymbol{c}}, \vec{\boldsymbol{d}} \leftarrow\$ \mathbb{Z}_p^{5N+2}$
$A_2 = F^{r_{A,2}} \cdot \vec{\boldsymbol{P}}^{(\xi\|\vec{\boldsymbol{r}}\|\vec{1}^N\|\vec{\boldsymbol{e}}_0\|sk\cdot\vec{\boldsymbol{e}}_0\|r\cdot\vec{\boldsymbol{e}}_0\|sk)} \cdot \vec{\boldsymbol{h}}_0^{(0\|\vec{\boldsymbol{e}}_0\|\vec{0}^N\|\vec{\boldsymbol{sk}}\|\vec{1}^N\|\vec{1}^N\|0)},$
$A_3 = F^{r_{A,3}} \cdot \vec{\boldsymbol{P}}^{\vec{\boldsymbol{c}}} \cdot \vec{\boldsymbol{h}}_0^{\vec{\boldsymbol{d}}}$
$\mathcal{P} \rightarrow \mathcal{V} : A_2, A_3$
$\mathcal{V}$ computes:
$e \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : e$
$\mathcal{P}$ computes:
$\theta_1 = r_{A,3} + e \cdot r_{A,2}, \boldsymbol{\theta}_2 = \boldsymbol{c}\|\boldsymbol{d} + e \cdot (\xi\|\eta\|1\|\hat{\boldsymbol{e}}\|0\|\vec{\boldsymbol{e}}_0\|\vec{0}^N\|\vec{sk}\|\vec{1}^N\|\vec{1}^N\|0)$
$\mathcal{P} \rightarrow \mathcal{V} : \theta_1, \boldsymbol{\theta}_2$
$\mathcal{V}$: Verification Procedure:
$F^{\theta_1} \cdot (\boldsymbol{P}\|\vec{h}_0)^{\boldsymbol{\theta}_2} \stackrel{?}{=} A_3 \cdot A_2^e$

.....

If verification passed, both prover and verifier compute
and verifier compute $A = A_1 \cdot A_2 \cdot (\boldsymbol{h}_1\|\boldsymbol{h}_2\|\boldsymbol{h}_3\|\boldsymbol{h}_4)^{-\vec{1}^{2N+2\beta}} \cdot \boldsymbol{h}_5^{\vec{1}^N}$
$\mathcal{V}$ computes:
$w \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : w$
$\mathcal{P}$ computes:
$r_S \leftarrow\$ \mathbb{Z}_p, \vec{\boldsymbol{s}}_L \leftarrow\$ \mathbb{Z}_p^N, \vec{\boldsymbol{s}}_R \leftarrow\$ \mathbb{Z}_p^N : \forall i \in [m], \vec{\boldsymbol{c}}_R[i] = 0 \Rightarrow s_i = 0$
$S := F^{r_S} \vec{\boldsymbol{H}}^{\vec{\boldsymbol{s}}_L} \vec{\boldsymbol{G}}_w^{\vec{\boldsymbol{s}}_R}$
$\mathcal{P} \rightarrow \mathcal{V} : S$
$\mathcal{V}$ computes: $y, z \leftarrow\$ \mathbb{Z}_p$
$\mathcal{V} \rightarrow \mathcal{P} : y, z$
$\mathcal{P}$ computes:
Define the following polynomials (in X) :
$l(X) := \vec{\boldsymbol{c}}_L + \vec{\alpha} + \vec{\boldsymbol{s}}_L \cdot X, r(X) := \theta \circ (\vec{\boldsymbol{c}}_R + \vec{\boldsymbol{s}}_R \cdot X) + \vec{\mu},$
$t(X) :=< l(X), r(X) >= t_2 \cdot X^2 + t_1 \cdot X + t_0$ and $t_0 = \delta$
$\tau_1, \tau_2 \leftarrow\$ \mathbb{Z}_p, T_1 = g^{t_1} F^{\tau_1}, T_2 = g^{t_2} F^{\tau_2} \mathcal{P} \rightarrow \mathcal{V} : T_1, T_2$
$\mathcal{P}$ computes:
$\tau = \tau_1 x + \tau_2 x^2, r = r_A + r_S \cdot x, (\vec{l}, \vec{r}, t) = (l(x), r(x), t)$
$\mathcal{P} \rightarrow \mathcal{V} : \tau, r, \vec{l}, \vec{r}, t$
$\mathcal{V}$ check if the following relations hold:
$t \stackrel{?}{=}< \vec{l}, \vec{r} >, F^r \vec{\boldsymbol{G}}_w^{\vec{l}} \vec{\boldsymbol{H}}^{\vec{\theta}^{\circ -1} \circ \vec{r}} \stackrel{?}{=} A S^x \vec{\boldsymbol{G}}_w^{\vec{\alpha}} \vec{\boldsymbol{H}}^{\vec{\beta}}, G^t F^\tau \stackrel{?}{=} G^\delta T_1^x T_2^x$

Protocol 4: The protocol of Anonymous Zether.

Meanwhile, since sk' is generated randomly, U' does not exist in the former key image set.

c) Since B_1, A, S_1, S_3, T_1, T_2, τ_x, μ, z_{α_1}, z_{α_2}, $\hat{t}$, $\boldsymbol{l}$ and $\boldsymbol{r}$ are generated as RingCT-3.0, the proof can pass the other equalities of the verification procedure. Hence, the "forged" proof can pass the verification procedure without knowing any secret key of $\boldsymbol{vk}$ and any opening of $\boldsymbol{c}$.

By the analysis above, one can forge a proof and spend any coin of any amount he wants, and the verification procedure would never catch him. For the case of multiple input coins and output coins, we need to adjust the construction of B_2 and make sure that all the legal verification keys and all the legal commitments can be eliminated from B_1, and the verification keys and the commitments of the "forged" coins can be appended into B_1.

Appendix C. Security Proof of Theorem 1

We give the security proof of Theorem 1 as follows.

Proof. 1) **Perfect completeness.** Perfect completeness follows directly.

2) **Perfect special honest-verifier zero-knowledge.** Given all the randomness (y, z, x) from the adversary, we construct a simulator that produces one proof $(S, T_1, T_2, \tau_x, \mu, \hat{t}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{\eta})$ without the witness $(\boldsymbol{b}_L, \boldsymbol{b}_R, \boldsymbol{a}, \rho_L)$. The distribution of the "produced" proof is indistinguishable from a valid proof.
The simulator randomly chooses $(T_2, \tau_x, \mu, \hat{t}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{\eta})$ from their respective domain and then calculates S and T_1 according to the verification equations in Protocol 1 as follows:

$$S = (h^\mu \cdot \boldsymbol{g}^{\boldsymbol{l}+z\cdot\mathbf{1}^{n_1}} \cdot \boldsymbol{g}_2^{\boldsymbol{\eta}} \cdot (\boldsymbol{h}')^{\boldsymbol{r}-z\cdot\boldsymbol{\alpha}\circ\boldsymbol{y}^{n_1} - \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta}_i} \tilde{P}^{-1})^{x^{-1}}$$
$$T_1 = (g^{\hat{t}-\delta(y,z)} \cdot h^{\tau_x} \cdot T_2^{-x^2})^{x^{-1}}$$

All proof components $(S, T_1, T_2, \tau_x, \mu, \hat{t}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{\eta})$ are chosen randomly or deduced by the verification equations, the distribution is identical to a real one. This protocol has perfect special honest-verifier zero-knowledge since the running time of this simulation is quite efficient, the distribution is the same as a real one, and the "produced" proof can pass the verification procedure.

3) **Computational witness-extended emulation.** To prove the witness-extended emulation, we construct an extractor $\chi_{\texttt{inner}}$ that rewinds the protocol $3 \cdot k \cdot n_1$ times for different (x, y, z) and deduces $3 \cdot k \cdot n_1$ different transcripts.

- From the second and third verification equations in Protocol 1, one can deduce:

$$\tilde{P} \cdot S^x = h^\mu \cdot \boldsymbol{g}^{\boldsymbol{l}+z\cdot\mathbf{1}^{n_1}} \cdot \boldsymbol{g}_2^{\boldsymbol{\eta}} \cdot (\boldsymbol{h}')^{\boldsymbol{r}-z\cdot\boldsymbol{\alpha}\circ\boldsymbol{y}^{n_1} - \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta}_i} \tag{36}$$

For given values of (y, z), using two valid transcripts for different x challenges x_1 and x_2, one can compute

ξ_1 and ξ_2 such that $\xi_1 + \xi_2 = 1$ and $x_1\xi_1 + x_2\xi_2 = 0$. By equation (36), we have

$$\tilde{P} = h^{\xi_1\mu_1+\xi_2\mu_2} \cdot \boldsymbol{g}^{\xi_1\cdot\boldsymbol{l}_1+\xi_2\cdot\boldsymbol{l}_2+z\cdot\mathbf{1}^{n_1}} \cdot \boldsymbol{g}_2^{\xi_1\cdot\boldsymbol{\eta}_1+\xi_2\cdot\boldsymbol{\eta}_2} \cdot (\boldsymbol{h}')^{\xi_1\cdot\boldsymbol{r}_1+\xi_2\cdot\boldsymbol{r}_2-z\cdot\boldsymbol{\alpha}\circ\boldsymbol{y}^{n_1}-\sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta}_i} \quad (37)$$

Let $\tilde{P} = h^{\rho'_L}\boldsymbol{g}^{\boldsymbol{b}'_L}\boldsymbol{g}_2^{\boldsymbol{a}'}\boldsymbol{h}^{\boldsymbol{b}'_R}$ with unknown variables $\rho'_L \in \mathbb{Z}_p$, $\boldsymbol{b}'_L, \boldsymbol{b}'_R \in \mathbb{Z}_p^{n_1}$, and $\boldsymbol{a}' \in \mathbb{Z}_p^{n_2}$, and then we can deduce the following equations from (37).

$$\begin{cases} \rho'_L = \xi_1\mu_1 + \xi_2\mu_2 \\ \boldsymbol{b}'_L = \xi_1\cdot\boldsymbol{l}_1 + \xi_2\cdot\boldsymbol{l}_2 + z\cdot\mathbf{1}^{n_1} \\ \boldsymbol{b}'_R = (\xi_1\cdot\boldsymbol{r}_1 + \xi_2\cdot\boldsymbol{r}_2)\circ\boldsymbol{y}^{-n_1} - z\cdot\boldsymbol{\alpha} \\ \qquad\quad - \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta}_i\circ\boldsymbol{y}^{-n_1} \\ \boldsymbol{a}' = \xi_1\cdot\boldsymbol{\eta}_1 + \xi_2\cdot\boldsymbol{\eta}_2 \end{cases} \quad (38)$$

Meanwhile, for given values of (y, z), using two valid transcripts for different x challenges x_1 and x_2, one can compute ξ'_1 and ξ'_2 such that $\xi'_1 + \xi'_2 = 0$ and $x_1\xi'_1 + x_2\xi'_2 = 1$. Then, one can deduce the only value $\rho'_S \in \mathbb{Z}_p$, $\boldsymbol{s}'_L, \boldsymbol{s}'_R \in \mathbb{Z}_p^{n_1}$, and $\boldsymbol{s}'_M \in \mathbb{Z}_p^{n_2}$ such that

$$\begin{aligned} S =& h^{\xi'_1\mu_1+\xi'_2\mu_2} \cdot \boldsymbol{g}^{\xi'_1\cdot\boldsymbol{l}_1+\xi'_2\cdot\boldsymbol{l}_2} \cdot \boldsymbol{g}_2^{\xi'_1\cdot\boldsymbol{\eta}_1+\xi'_2\cdot\boldsymbol{\eta}_2} \cdot \boldsymbol{h}'^{\xi'_1\cdot\boldsymbol{r}_1+\xi'_2\cdot\boldsymbol{r}_2} \\ =& h^{\rho'_S}\boldsymbol{g}_1^{\boldsymbol{s}'_L}\boldsymbol{g}_2^{\boldsymbol{s}'_M}\boldsymbol{h}^{\boldsymbol{s}'_R} \end{aligned} \quad (39)$$

Suppose for any other set of challenges (y, z, x), the extractor can compute a different representation of $\tilde{P}$ and S. Then, this yields a non-trivial DL relation among generators h, $\boldsymbol{g}$, $\boldsymbol{g}_2$ and $\boldsymbol{h}$, which contradicts the DL relation assumption.

Combining with equation (36) and the representations of $\tilde{P}$ and S, one can deduce that for all challenges (y, z, x):

$$\boldsymbol{l} = \boldsymbol{b}'_L - z\cdot\mathbf{1}^{n_1} + x\cdot\boldsymbol{s}'_L \quad (40)$$

$$\boldsymbol{r} = \boldsymbol{y}^{n_1}\circ(\boldsymbol{b}'_R + z\cdot\boldsymbol{\alpha} + x\cdot\boldsymbol{s}'_R) + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta_i} \quad (41)$$

$$\boldsymbol{\eta} = \boldsymbol{a}' + x\cdot\boldsymbol{s}'_M \quad (42)$$

Suppose these equalities do not hold for all challenges and $(\boldsymbol{l}, \boldsymbol{r})$ from the transcript. In that case, we have two distinct representations of the same group element using generators $h, \boldsymbol{g}, \boldsymbol{g}_2$, and $\boldsymbol{h}$. This would be a non-trivial DL relation.

- From the first verification equation in Protocol 1, one can deduce that

$$g^{\hat{t}-\delta(y,z)}h^{\tau_x} = T_1^x \cdot T_2^{x^2} \quad (43)$$

For given values of (y, z), using three valid transcripts of different x challenges x_1, x_2, and x_3, one can deduce that:

$$g^{\hat{t}_1-\hat{t}_2}h^{\tau_{x_1}-\tau_{x_2}} = T_1^{x_1-x_2} \cdot T_2^{x_1^2-x_2^2} \quad (44)$$

$$g^{\hat{t}_3-\hat{t}_2}h^{\tau_{x_3}-\tau_{x_2}} = T_1^{x_3-x_2} \cdot T_2^{x_3^2-x_2^2} \quad (45)$$

where $(\hat{t}_1, \tau_{x_1})$, $(\hat{t}_2, \tau_{x_2})$, and $(\hat{t}_3, \tau_{x_3})$ is the transcript corresponding to x_1, x_2, and x_3, respectively. Using linear combinations of (44) and (45), one can deduce the only values $t'_1, t'_2, \tau'_1, \tau'_2$, s.t., $T_1 = g^{t'_1}h^{\tau'_1}$ and $T_2 = g^{t'_2}h^{\tau'_2}$.

Substituting the representations of T_1 and T_2 into (43), we can deduce that

$$\hat{t} = \delta(y, z) + t'_1x + t'_2x^2 \quad (46)$$

Otherwise, this will yield a non-trivial DL relation between g and h.

Let $p(X) = \langle l(X), r(X)\rangle$ and $t(X) = t'_0 + t'_1X + t'_2X^2$, for all (y, z) challenges and three distinct challenges $X = x_j$, $j \in [1, 3]$:

$$t'_0 + t'_1X + t'_2X^2 - p(X) = 0$$

with $t'_0 = \delta(y, z)$ and $p(X) = p_0 + p_1X + p_2X^2 = \langle l(X), r(X)\rangle$. Since $t(X) - p(X)$ is of degree 2, but has at least three roots x_1, x_2, x_3, and then we have $t(X) - p(X) = 0$, i.e., $t(X) = \langle l(X), r(X)\rangle$. Therefore, $p_0 = t'_0$. By equations (40) and (41), we have for all y, z challenges:

$$\begin{aligned} \langle\boldsymbol{l}, \boldsymbol{r}\rangle =& \langle\boldsymbol{b}'_L - z\cdot\mathbf{1}^{n_1} + x\cdot\boldsymbol{s}'_L, \boldsymbol{y}^{n_1}\circ(\boldsymbol{b}'_R + z\cdot\boldsymbol{\alpha} \\ &+ x\cdot\boldsymbol{s}'_R) + \sum_{i=1}^{k} z^{i+1}\cdot\boldsymbol{\zeta_i}\rangle \\ =& -\sum_{i=1}^{k} z^{i+2}\cdot\langle\boldsymbol{\zeta_i}, \mathbf{1}^{n_1}\rangle + \sum_{i=1}^{k} z^{i+1}\cdot\langle\boldsymbol{b}'_L, \boldsymbol{\zeta_i}\rangle \\ &- z^2\cdot\langle\boldsymbol{\alpha}, \boldsymbol{y}^{n_1}\rangle + z\cdot[\langle\boldsymbol{\alpha}\circ\boldsymbol{b}'_L, \boldsymbol{y}^{n_1}\rangle \\ &- \langle\boldsymbol{b}'_R, \boldsymbol{y}^{n_1}\rangle] + \langle\boldsymbol{b}'_L, \boldsymbol{y}^{n_1}\circ\boldsymbol{b}'_R\rangle + t'_1x + t'_2x^2 \\ =& \delta(y, z) + t'_1x + t'_2x^2 \end{aligned} \quad (47)$$

Then, (47) is equivalent to:

$$\sum_{i=1}^{k} z^{i+1}\cdot[\langle\boldsymbol{b}'_L, \boldsymbol{\zeta_i}\rangle - d_i] + z\cdot\langle\boldsymbol{\alpha}\circ\boldsymbol{b}'_L - \boldsymbol{b}'_R + \boldsymbol{\beta}, \boldsymbol{y}^{n_1}\rangle + \langle\boldsymbol{b}'_L\circ\boldsymbol{b}'_R + \boldsymbol{\gamma}, \boldsymbol{y}^{n_1}\rangle = 0 \quad (48)$$

If this equation holds for n_1 distinct y challenges and k distinct z challenges, one can infer that $\boldsymbol{b}'_L \circ \boldsymbol{b}'_R = -\boldsymbol{\gamma}$, $\boldsymbol{\alpha}\circ\boldsymbol{b}'_L - \boldsymbol{b}'_R + \boldsymbol{\beta} = \mathbf{0}^{n_1}$, and $\langle\boldsymbol{b}'_L, \boldsymbol{\zeta_i}\rangle = d_i$ for $1 \le i \le k$.

Meanwhile, from (38), one can deduce the only value $\boldsymbol{a}'$ that satisfies $\tilde{P} = h^{\rho'_L}\boldsymbol{g}^{\boldsymbol{b}'_L}\boldsymbol{g}_2^{\boldsymbol{a}'}\boldsymbol{h}^{\boldsymbol{b}'_R}$. Otherwise, there is a non-trivial DL relation. Hence, the extracted value $(\boldsymbol{b}'_L, \boldsymbol{b}'_R, \boldsymbol{a}', \rho'_L)$ is a valid witness for relation $\tilde{\mathcal{R}}_{\texttt{inner}}$ of (7). Meanwhile, the extracted value is also a valid witness for relation $\mathcal{R}_{\texttt{inner}}$ of (2) since $\tilde{\mathcal{R}}_{\texttt{inner}}$ and $\mathcal{R}_{\texttt{inner}}$ are equivalent.

The extractor rewinds the prover $3 \cdot k \cdot n_1$ times in total. The extraction is very efficient, and the number of transcripts is a polynomial of the security parameter.

□

Appendix D.
Applications: Ring Signatures

Ring signatures allow a signer to dynamically choose a set of verification keys $\boldsymbol{vk} = (vk_1, \cdots, vk_N)$ (including his own) and sign the message on behalf of the set without revealing his identity. The verifier can ensure that one user in the verification key set $\boldsymbol{vk}$ signs the message without knowing who signs it.

A ring signature scheme consists of four PPT algorithms (Setup, UKGen, Sign, Vfy) for generating public parameters available to all users, generating keys for users, signing messages, and verifying ring signatures.

Next, we will give an SHVZK protocol for the one-out-of-N proof of knowledge using the model of Section 3, i.e., a zero-knowledge protocol $\Pi_{1\text{-}N}$ for the following relation:

$$\mathcal{R}_{1\text{-}N} = \{(g, \boldsymbol{vk}) | \exists\ vk, sk, \text{ s.t., } vk \in \boldsymbol{vk} \wedge vk = g^{sk}\}, \quad (49)$$

where $\boldsymbol{vk} = (vk_1, \cdots, vk_N) \in \mathbb{G}^N$, $g, vk \in \mathbb{G}$ and $sk \in \mathbb{Z}_p$.

D.1. One-out-of-N Proof of Knowledge

In this section, we construct an SHVZK protocol for the one-out-of-N proof of knowledge using the model of Section 3, i.e., one can deduce $\mathcal{R}_{1\text{-}N}$ of (49) from $\mathcal{R}_{\texttt{KKDL}}$ of (1) as follows.

1) Let $m = 1$, $n_1 = N$, $n_2 = 1$, $k = 1$, $f(\boldsymbol{b}_L) = \boldsymbol{b}_L \circ \boldsymbol{b}_L - \boldsymbol{b}_L = \boldsymbol{0}^N$, $\phi_1(\boldsymbol{b}_L) = \sum_{i=1}^{N} b_{L,i}$, $d_1 = 1$, $\boldsymbol{q}_{1,1} = \boldsymbol{vk} = \{vk_1, \cdots, vk_N\}$, $\boldsymbol{q}_2 = \{g\}$ and $\boldsymbol{\psi}_1 = \{-sk\}$ in $\mathcal{R}_{\texttt{KKDL}}$.
2) Since $f(\boldsymbol{b}_L) = \boldsymbol{b}_L \circ \boldsymbol{b}_L - \boldsymbol{b}_L = \boldsymbol{b}_L \circ (\boldsymbol{b}_L - \boldsymbol{1}^N) = \boldsymbol{0}^N$, each component of $\boldsymbol{b}_L$ is either 0 or 1. Since $\phi_1(\boldsymbol{b}_L) = \sum_{i=1}^{N} b_{L,i} = 1$, $\boldsymbol{b}_L$ has exactly one component that is "1". Meanwhile, $\boldsymbol{q}_{1,1}^{\boldsymbol{b}_L} \cdot \boldsymbol{q}_2^{\boldsymbol{\psi}_1} = \boldsymbol{vk}^{\boldsymbol{b}_L} \cdot g^{-sk} = 1$ means that there exists $vk \in \boldsymbol{vk}$ and $sk \in \mathbb{Z}_p$ such that $vk = g^{sk}$, so $\mathcal{R}_{1\text{-}N}$ can be deduced directly.

Therefore, one can use $\Pi_{\texttt{KKDL}}$ and $\Pi_{\texttt{inner}}$ to construct a zero-knowledge protocol $\Pi_{1\text{-}N}$ for $\mathcal{R}_{1\text{-}N}$. Since $m = 1$ and $n_2 = 1$, the in-line Σ-protocol of Protocol 2 is no longer needed. At the beginning of the protocol, the prover commits $(-sk, \boldsymbol{b}_L)$ and sends $P = h^{\rho} \cdot \hat{\boldsymbol{g}}_1^{\boldsymbol{b}_L} \cdot \hat{g}_2^{-sk}$ to the verifier, where $\rho \leftarrow_{\$} \mathbb{Z}_p$. Secondly, the verifier chooses $e \leftarrow_{\$} \mathbb{Z}_p$ and sends it to the prover. After that, the prover and verifier proceed to $\Pi_{\texttt{inner}}$ with $\boldsymbol{g}_1 = \hat{\boldsymbol{g}}_1 \circ \boldsymbol{vk}^e$ and $g_2 = \hat{g}_2 \circ g^e$.

The security of $\Pi_{1\text{-}N}$ is shown in Theorem 4. The total proof size of $\Pi_{1\text{-}N}$ is $2 \cdot \lceil \log_2(N) \rceil + 10$ using the improved inner-product argument and the Fiat-Shamir heuristic (since $n_2 = 1$ and $|\boldsymbol{\eta}| = 1$, we do not use the improved inner-product argument to $\boldsymbol{\eta}$ of Protocol 1).

Theorem 4 (Security of $\Pi_{1\text{-}N}$)**.** *The protocol $\Pi_{1\text{-}N}$ has perfect completeness, perfect special honest-verifier zero-knowledge, and computational witness-extended emulation.*

The proof of Theorem 4 can be deduced from the proofs of Theorem 2 and Theorem 1.

The above protocol can be converted to a signature of knowledge (SoK) [39] which implies a ring signature scheme following the methods of [31].

Prover		**Verifier**
		$d_1, d_2 \leftarrow\$ \mathbb{Z}_p$
	$\xleftarrow{d_1, d_2}$	
	Both prover and verifier compute $\boldsymbol{g} \leftarrow \boldsymbol{vk} \circ \boldsymbol{c}^{d_1} \circ \hat{\boldsymbol{g}}^{d_2}$	
Compute: $\alpha_1, \alpha_2, \beta, \rho, r_{\alpha_1}, r_{\alpha_2}, r_{sk}, r_\Delta \leftarrow\$ \mathbb{Z}_p$ $\boldsymbol{s}_L, \boldsymbol{s}_R \leftarrow\$ \mathbb{Z}_p^N$ $B_1 = h^{\alpha_1} \boldsymbol{g}^{\boldsymbol{b}_L} = h^{\alpha_1} \cdot vk_{\text{ind}} \cdot c_{\text{ind}}^{d_1} \cdot \hat{g}_{\text{ind}}^{d_2}$ $A = h^\beta \cdot \boldsymbol{h}^{\boldsymbol{b}_R}, S_2 = h^\rho \boldsymbol{g}^{\boldsymbol{s}_L} \boldsymbol{h}^{\boldsymbol{s}_R}$ $S_1 = h^{r_{\alpha_1} - d_2 r_{\alpha_2}} g^{r_{sk}} \cdot g_c^{d_1 r_\Delta}$, $B_2 = h^{\alpha_2} \cdot \hat{g}_{\text{ind}}, S_3 = U^{r_{sk}}$ $\boxed{B_2' = h^{\alpha_2} \cdot \hat{g}_{\text{ind}} \cdot (vk_{\texttt{ind}} \cdot vk'^{-1})^{\frac{1}{d_2}} \cdot (c_{\texttt{ind}} \cdot c'^{-1})^{\frac{d_1}{d_2}}}$ $\boxed{S_3' = U'^{r_{sk}}}$		
	$\xrightarrow{B_1, B_2, A, S_1, S_2, S_3}$	
	$\boxed{\xrightarrow{B_1, B_2', A, S_1, S_2, S_3'}}$	
		$y, z, w \leftarrow\$ \mathbb{Z}_p$
	$\xleftarrow{y, z, w}$	
Compute: $l(X) = \boldsymbol{b}_L - z \cdot \mathbf{1}^N + \boldsymbol{s}_L \cdot X$ $r(X) = \boldsymbol{y}^N \circ (w \cdot \boldsymbol{b}_R + wz \cdot \mathbf{1}^N + \boldsymbol{s}_R \cdot X) + z^2 \cdot \mathbf{1}^N$ $t(X) = \langle l(X), r(X) \rangle = t_0 + t_1 X + t_2 X^2$ $\tau_1, \tau_2 \leftarrow\$ \mathbb{Z}_p, T_1 = g^{t_1} h^{\tau_1}, T_2 = g^{t_2} h^{\tau_2}$		
	$\xrightarrow{T_1, T_2}$	
		$x \leftarrow\$ \mathbb{Z}_p$
	$\xleftarrow{x}$	
Compute: $\boldsymbol{l} = l(x) = \boldsymbol{b}_L - z \cdot \mathbf{1}^N + x \cdot \boldsymbol{s}_L$ $\boldsymbol{r} = r(x) = \boldsymbol{y}^N \circ (w \cdot \boldsymbol{b}_R + wz \cdot \mathbf{1}^N + x \cdot \boldsymbol{s}_R) + z^2 \cdot \mathbf{1}^N$ $\hat{t} = \langle \boldsymbol{l}, \boldsymbol{r} \rangle$ $\tau_x = \tau_2 \cdot x^2 + \tau_1 \cdot x, \mu = \alpha_1 + \beta \cdot w + \rho \cdot x,$ $z_{\alpha_1} = r_{\alpha_1} + \alpha_1 \cdot x, z_{\alpha_2} = r_{\alpha_2} + \alpha_2 \cdot x,$ $z_{sk} = r_{sk} + sk \cdot x, z_\Delta = r_\Delta + \Delta \cdot x$ $\boxed{z_{sk}' = r_{sk} + sk' \cdot x, z_\Delta' = r_\Delta + \Delta' \cdot x}$		
	$\xrightarrow{\tau_x, \mu, z_{\alpha_1}, z_{\alpha_2}, z_{sk}, z_\Delta, \hat{t}, \boldsymbol{l}, \boldsymbol{r}}$	
	$\boxed{\xrightarrow{\tau_x, \mu, z_{\alpha_1}, z_{\alpha_2}, z_{sk}', z_\Delta', \hat{t}, \boldsymbol{l}, \boldsymbol{r}}}$	
		Verification procedure: Let $\boldsymbol{h}' = \boldsymbol{h}^{\circ \boldsymbol{y}^{-N}}$, $g^{\hat{t} - \delta(y,z,w)} h^{\tau_x} \stackrel{?}{=} T_1^x \cdot T_2^{x^2}$ Let $T = B_1 \cdot A^w \cdot S_2^x \cdot \boldsymbol{g}^{-z \cdot \mathbf{1}^N} \cdot (\boldsymbol{h}')^{wz \cdot \boldsymbol{y}^N + z^2 \cdot \mathbf{1}^N}$ $T \stackrel{?}{=} h^\mu \cdot \boldsymbol{g}^{\boldsymbol{l}} \cdot (\boldsymbol{h}')^{\boldsymbol{r}}$ $\hat{t} \stackrel{?}{=} \langle \boldsymbol{l}, \boldsymbol{r} \rangle$ $h^{z_{\alpha_1} - d_2 z_{\alpha_2}} g^{z_{sk}} g_c^{d_1 z_\Delta} \stackrel{?}{=} S_1 (B_1 \cdot \tilde{c}^{-d_1} \cdot B_2^{-d_2})^x$ $U^{z_{sk}} \stackrel{?}{=} S_3 \cdot u^x$ $\boxed{h^{z_{\alpha_1} - d_2 z_{\alpha_2}} g^{z_{sk}'} g_c^{d_1 z_\Delta'} \stackrel{?}{=} S_1 \cdot (B_1 \cdot (\tilde{c}')^{-d_1} \cdot B_2'^{-d_2})^x}$ $U'^{z_{sk}'} \stackrel{?}{=} S_3' \cdot u^x$

Protocol 5: The RingCT-3.0 and the cryptanalysis of RingCT-3.0 (the cryptanalysis procedures are shown in the box that replaces the corresponding steps of the original protocol).