Revisiting Linkable Ring Signatures with Logarithmic Verification Complexity

Danai Balla[1,2] and Pyrros Chaidos[3,4,5]

[1] National Technical University of Athens
[2] Archimedes, Athena Research Center, Greece
[3] National and Kapodistrian University of Athens
[4] IOG, Greece
[5] Common Prefix, Greece

Abstract. Ring Signatures allow a user to sign on behalf of an ad-hoc set of public keys, while hiding their identity inside that set. Linkable Ring Signatures (LRS) add the functionality of detecting signatures originating from the same signer. They have found many applications in anonymous transactions and e-voting.

The `LLRing` family of linkable ring signature schemes by Hui and Chau (ESORICS 2024) is one of the more efficient LRS schemes. However, we show that it has an unlinkability vulnerability, meaning an adversary can create more unlinkable signatures than the number of secret keys they own. The vulnerability is caused by the introduction of unwanted structure to base elements used in proofs. We also find a similar attack against the Threshold Ring Referral (TRR) scheme of Ta, Hui, and Chau (Security and Privacy 2025), rendering it unsound.

We show how to achieve strong linkability with logarithmic verification complexity in the pairing based setting by first reverting the unsafe construction of base elements, and by also adjusting the arguments of knowledge used in order to maintain efficiency. We separate the design and analysis of the scheme from the instantiation of the knowledge arguments, which helps prevent unwanted interactions between the two, and can provide easier upgrades to more efficient proof systems.

Keywords: Digital Signatures · Zero Knowledge Proof Systems · Cryptographic Attacks

1 Introduction

Ring Signatures [23] allow a user to sign on behalf of a set of public keys, while hiding their identity inside that set, and without the need for an administrator. Linkable Ring Signatures (LRS) [21] extend ring signatures by adding the functionality of detecting signatures originating from the same signer. They have

Errata Previous versions of this work contained an error in the construction of Section 4. This version corrects this mistake by altering the protocol and adding additional constraints to the verification equations. See the Errata section for details.

found many applications in anonymous transactions [22,17,24,31] and electronic voting [21,26,13,10].

When using (L)RS schemes with large rings it is important for performance that the signature size scales sublinearly with the size of the ring, and for this reason there have been many works constructing schemes that achieve logarithmic size [14,30,17,31,15,7,28] w.r.t. the number of ring members, or even constant size when a trusted setup is used [26,2,27]. Apart from the signature size, another important parameter is the verification complexity. Recently, Bootle et al. [7] proposed linkable ring signatures with logarithmic verification complexity and transparent setup, constructing the scheme DualDory. DualDory adds linkability to the DualRing [30] RS scheme and achieves logarithmic verification complexity by having the verifier perform a ring-dependent linear time preproccesing that can later be used to verify signatures in logarithmic time for the same ring.

Linkability Vulnerabilities in LRS schemes Hui and Chau [15] discovered a vulnerability in the linkability of DualDory which results in an adversary knowing two secret keys being able to create multiple unlinkable signatures, contradicting its security guarantees. In the same work they propose the `LLRing` schemes, which are based on Omniring [17] and claim to eliminate such an attack. They propose the DLOG-based scheme `LLRing-DL` that improves on the verification complexity of Omniring, and `LLRing-P`, a pairing-based LRS scheme with logarithmic verification complexity.

In this work, we show that the linkability of the `LLRing` schemes is also vulnerable to an efficient attack, contradicting their security guarantees. The inspiration for the attack came from a recent paper [19] of Lin et al. which points out that if a prover already knows some discrete logarithm relations between the generators used to construct the non interactive zero knowledge proofs (NIZKs – specifically Bulletproofs) used in the construction, they can create invalid proofs. This observation also applies to both variants of the `LLRing` protocol and leads to the unlinkability attack we present. The more general insight of course is that the knowledge soundness (or equivalent) property of many NIZKs is achieved via the existence of an extractor who can produce *either* a satisfying witness *or* a non-trivial discrete log relation between base elements. Therefore, if such relations are a-priori known to the users, knowledge soundness may be vacuous: it is possible that the extractor will simply rediscover a relation that was already known. After understanding the cause of the attack, we propose a linkable ring signature scheme with logarithmic verification complexity and strong linkability inspired by `LLRing`.

Soundness Vulnerability in Ring Referral Another line of work [29,25] considers the scenario of producing functionality similar to that of a ring signature from a signature generated prior to defining a ring. More concretely, a user receives a signature from a signer and then proves knowledge of a signature originating from an anonymity set of public keys that includes the signer. Both schemes use an Omniring-style protocol to achieve this; the user first commits to the public key and to the signature, then shows that the committed signature verifies with

respect to the committed public key and then proves that the committed public key belongs to the anonymity set.

One of these works, Ring Referral [25], uses a protocol similar to `LLRing-P` to show inclusion of the committed public key in the anonymity set. We find that the threshold version of the Ring Referral protocol [25] suffers from a soundness attack. In Threshold Ring Referral, a user proves that they own k signatures originating from k distinct signers. We show that a user collaborating with $m < k$ signers can construct a proof that passes verification while having signatures originating only from those m signers.

1.1 Our contributions

Our contributions can be summarized as follows.

- We show that the `LLRing` [15] LRS scheme has a unlinkability vulnerability, meaning an adversary who knows multiple secret keys can create multiple unlinkable signatures (more than the number of secret keys they own).
- We also show that a similar attack applies to the Threshold Ring Referral scheme of Ta, Hui and Chau [25].
- We construct the first pairing-based LRS scheme with strong linkability and logarithmic verification complexity.
- We also restore the linkability of `LLRing-DL` by constructing a modification to the Bulletproofs [9] protocol.

1.2 Related work

Ring signatures and Linkable Ring signatures Ring signatures were first proposed by Rivest et al. [23] albeit with size linear in the ring size. Groth and Kohlweiss constructed a ring signature scheme with logarithmic size but $O(n \log n)$ prover complexity using a custom zero knowledge protocol. Later constructions made use of advances in zero knowledge proof systems. One example is DualRing [30] by Yuen et al, a ring signature scheme with logarithmic size and linear prover time, achieved by compressing the signature using Bulletproofs [9]. Linkable Ring Signatures were first introduced by Liu et al. [21], allowing a user to hide their anonymity inside a ring whilst ensuring that signatures by the same signer can be linked. RingCT 3.0 [31] and Omniring [17] were proposed in the context of confidential transactions and achieve a logarithmic signature size and linear prover time, using a modification of the range proof protocol of Bulletproofs.

Linkable Ring Signatures with logarithmic verification complexity where first proposed by Bootle et al. [7], were they proposed DualDory, adding linkability to the DualRing construction and replacing Bulletproofs with Dory [18]. In their work [15], Hui and Chau identified an attack on the linkability of DualDory and proposed the schemes `LLRing-DL` and `LLRing-P`. `LLRing-DL` uses a modification of the range proof protocol of Bulletproofs and claims to achieve better verification complexity than Omniring through preproccesing (but still linear), and `LLRing-P` uses Dory to compress the signature size and achieve logarithmic verification

Scheme	Setting	Size	Verifier time		Prover time	Linkable
			Offline	Online		
LSAG [21]	DLOG	$O(n)$	–	$O(n)$	$O(n)$	yes
One-out-of-many [14]	DLOG	$O(\log n)$	–	$O(n)$	$O(n \log n)$	no
DualRing [30]	DLOG	$O(\log n)$	–	$O(n)$	$O(n)$	no
Omniring [17]	DLOG	$O(\log n)$	–	$O(n)$	$O(n)$	yes
Xue et al. [28]	CRHF	$O(\text{poly}\log(\log n))$	$O(n)$	$O(\text{poly}\log(\log n))$	quasilinear	yes
DualDory [7]	SXDH	$O(\log n)$	$O(n)$	$O(\log n)$	$O(n)$	weak[1]
`LLRing-DL` [15]	DLOG	$O(\log n)$	$O(n)$	$O(n)$	$O(n)$	weak[2]
`LLRing-P` [15]	SXDH	$O(\log n)$	$O(n)$	$O(\log n)$	$O(n)$	weak[2]
Ours	SXDH, GPI	$O(\log n)$	$O(n)$	$O(\log n)$	$O(n)$	yes

Table 1. Performance comparison of (L)RS schemes with transparent setup. [1] An attack on the linkability of DualDory has been identified [15].
[2] We identify an attack on the linkability of `LLRing-DL` and `LLRing-P`.

complexity. Lastly, Xue et al. use STARKs to construct a post-quantum Linkable Ring Signature scheme with polylogarithmic signature size and verification complexity.

Transparent Zero Knowledge Proof Systems As established above, there is a strong link between the design of modern LRS schemes and zero knowledge protocols, made especially relevant by the linkability vulnerabilities we discuss. `LLRing-DL` uses Bulletproofs whilst `LLRing-P` and Ring Referral use Dory. In both proof systems, security is based on assumptions of hardness w.r.t. finding non-trivial relations between given elements (cf. Theorems 1,3), however, the schemes construct said elements in a way that relations can be found efficiently in real-world situations. This negates any security guarantees by the proof systems.

Comparison Using a trusted setup, it is possible to construct constant size and verification complexity LRS schemes. Examples include RSA-based constructions [26,2] and the more recent powers-of-tau based construction [27]. However, due to the potential of ring signatures to be used in a more ad-hoc manner, we believe the transparent setting is the more relevant one. A comparison of Ring and Linkable Ring Signature schemes with transparent setup appears in Table 1. We note that as the scheme of Xue et. al. [28] is based on STARKS it is quantum resistant but that comes at the cost of higher size constants than other schemes. In addition, partially changing the ring requires more effort than discrete log and pairing based schemes.

Post-Quantum Linkable Ring Signatures The work of Beullens et al. [5] develops linkable ring signatures for Isogeny and Lattice-based settings which are presumed to be quantum resistant. Their scheme achieves logarithmic size, but requires linear verification time. Due to the settings, the constants involved hinder a direct comparison.

1.3 A note on strong and weak linkability

The (strong) linkability property of LRS schemes requires that an adversary can create as many unlinkable signatures as the number of secret keys they own. However, some schemes satisfy only a weaker notion of linkability, *weak linkability* (as identified by Choi et al. [10]), which considers only adversaries that posess only *one* secret key from the ring of public keys, and this adversary should not be able to construct more than one unlinkable signature. Weak linkability is insufficient in real-world scenarios since the assumption of a signer knowing more than one secret key is realistic, especially in the blockchain setting. Some LRS schemes in the literature satisfy only weak linkability [20,4], but there are also constructions that satisfy strong linkability, such as LSAG[21], Omniring [17].

The notion of *weak* linkability can also be applied in a retroactive way to schemes such as DualDory and LLRing. We believe their original security proofs can be restated with minimal additions to (correctly) prove weak linkability. In brief, the proofs fail because they assume the extractor always produces a valid witness. In both settings (weak and strong) this is not strictly the case: it may produce a non-trivial discrete log relation instead. In the strong setting, this relation may involve multiple adversarial keys, an event which is the basis for the attack. In the weak setting, there is only a single adversarial key, so a non-trivial relation will cause a contradiction.

1.4 Outline

Section 2 introduces the notation and definitions used throughout. Section 3 describes our attacks on the `LLRing` family of schemes. Section 4 introduces our suggested replacement for `LLRing-P`. Section 5 presents an attack on Threshold Ring Referral. Our work concludes with a discussion section.

2 Preliminaries

2.1 Notation

We use $\leftarrow\!\!\$$ to denote uniformly random selection from a set. We denote by $[n]$ the set $\{1, \ldots, n\}$. Number $q \in \mathbb{N}$ is always a prime number. Groups are denoted by $\mathbb{G}$ and the field of order q is denoted by $\mathbb{Z}_q$. We use $1_{\mathbb{G}}$ to denote the identity element of group $\mathbb{G}$ or 1 if the context is clear. The function $\mathsf{negl}(\lambda)$ is a negligible function in λ. We also use the word *prover* to refer to a signer.

For $\vec{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q^n, \vec{G} = (G_1, \ldots, G_n) \in \mathbb{G}^n$, denote $\vec{G}^{\vec{a}} = \prod_{i\in[n]} G_i^{a_i}$ and $\vec{G}^{\circ\vec{a}} = (G_1^{a_1}, \ldots, G_n^{a_n})$. For a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, $\vec{G} = (G_1, \ldots, G_n) \in \mathbb{G}_1^n$, $\vec{H} = (H_1, \ldots, H_n) \in \mathbb{G}_2^n$ and $\vec{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q^n$, denote $e(\vec{G}, \vec{H}) = \prod_{i\in[n]} e(G_i, H_i)$ and $e(\vec{G}, \vec{H})^{\vec{a}} = \prod_{i\in[n]} e(G_i, H_i)^{a_i}$. Finally, for an even number $n \in \mathbb{N}$ and $\vec{G} = (G_1, \ldots, G_n) \in \mathbb{G}^n$, denote $\vec{G}_L = (G_1, \ldots, G_{\frac{n}{2}}) \in \mathbb{G}^{\frac{n}{2}}$ and $\vec{G}_R = (G_{\frac{n}{2}+1}, \ldots, G_n) \in \mathbb{G}^{\frac{n}{2}}$.

2.2 Linkable Ring Signatures

Linkable Ring Signatures allow a user to hide their identity inside a ring, while guaranteeing that signatures originating from the same user for the same event identifier can be linked together. We follow the definition of LRS from [7], [15].

Definition 1 (Linkable Ring Signature). *A linkable ring signature (LRS) scheme is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Link})$ *with message space* $\mathcal{M}$ *and event space* $\mathcal{F}$ *where*

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ *produces public parameters* pp*, which we assume to be available to all the algorithms below.*
- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}()$ *produces a key pair.*
- $\sigma \leftarrow \mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}, m, f)$*, where* $m \in \mathcal{M}$ *is a message,* $f \in \mathcal{F}$ *is an event, and* $\vec{\mathsf{pk}}$ *is a vector of public keys produced by* KeyGen *that includes the public key* pk *corresponding to secret key* sk*.*
- $b \leftarrow \mathsf{Verify}(\vec{\mathsf{pk}}, m, f, \sigma)$ *outputs a bit* $b \in \{0, 1\}$ *corresponding to the validity of signature* σ *under message* $m \in \mathcal{M}$*, event* $f \in \mathcal{F}$ *and ring* $\vec{\mathsf{pk}}$*.*
- $b \leftarrow \mathsf{Link}(\sigma, \sigma', f)$ *outputs a bit* $b \in \{0, 1\}$ *that corresponds to whether signatures* σ, σ' *originate from the same signer.*

An LRS should satisfy *correctness*, *anonymity*, *unforgeability*, *linkability* and *non-slanderability*.

Correctness states that honestly generated signatures always verify, signatures generated from the same signer for the same event can always be linked, and that the probability that signatures originating from different signers for the same event can be linked is negligible.

Definition 2 (Correctness). *An LRS scheme satisfies correctness if for any* $m_1, m_2 \in \mathcal{M}$*, any event* $f \in \mathcal{F}$*, any* $(\mathsf{sk}_1, \mathsf{pk}_1), (\mathsf{sk}_2, \mathsf{pk}_2)$ *produced by* KeyGen *with* $\mathsf{pk}_1 \neq \mathsf{pk}_2$ *and any ring of public keys* $\vec{\mathsf{pk}}$ *produced by* KeyGen *such that* $\mathsf{pk}_1, \mathsf{pk}_2$ *are included in* $\vec{\mathsf{pk}}$*, the following hold.*

- $\mathsf{Verify}(\vec{\mathsf{pk}}, m_1, f, \mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}_1, m_1, f)) = 1.$
- $\mathsf{Link}(\mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}_1, m_1, f), \mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}_1, m_2, f), f) = 1$
- $\Pr[\mathsf{Link}(\mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}_1, m_1, f), \mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}_2, m_2, f), f) = 1] \leq \mathsf{negl}(\lambda).$

In what follows, we model adversarial capabilities with oracles.

Definition 3 (Corruption oracle). *When queried with a public key* pk *produced by* KeyGen*, the corruption oracle* $\mathcal{CO}$ *returns the corresponding secret key* sk*, and keeps* pk *in its memory.*

Definition 4 (Signing oracle). *For a ring of public keys* $\vec{\mathsf{pk}}$*, when queried with* $\mathsf{pk} \in \vec{\mathsf{pk}}$*, message* m *and event* f*, the signing oracle* $\mathcal{SO}_{\vec{\mathsf{pk}}}$ *returns a signature* σ *whose distribution is computationally indistinguishable from the output of* $\mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}, m, f)$*, where* sk *is the secret key corresponding to public key* pk*. The oracle keeps the tuple* $(\sigma, \mathsf{pk}, m, f)$ *in its memory.*

Game 1: Anonymity game $\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
for $i \in [n]$, $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}()$, $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
$(m, f, \mathsf{pk}_{j_0}, \mathsf{pk}_{j_1}) \leftarrow \mathcal{A}^{\mathcal{CO}, \mathcal{SO}_{\vec{\mathsf{pk}}}}(\mathsf{pp}, \vec{\mathsf{pk}})$
$b \leftarrow \{0, 1\}$
if $\{j_0, j_1\} \notin [n]$ **then**
  return 0
$\sigma \leftarrow \mathsf{Sign}(\vec{\mathsf{pk}}, \mathsf{sk}_{j_b}, m, f)$
$b' \leftarrow \mathcal{A}(\sigma)$
return
$(b = b' \wedge \mathsf{pk}_{j_0} \notin \mathcal{CO} \wedge \mathsf{pk}_{j_1} \notin \mathcal{CO} \wedge (\cdot, \mathsf{pk}_{j_0}, \cdot, f) \notin \mathcal{SO}_{\vec{\mathsf{pk}}} \wedge (\cdot, \mathsf{pk}_{j_1}, \cdot, f) \notin \mathcal{SO}_{\vec{\mathsf{pk}}})$

Game 2: Unforgeability game $\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
for $i \in [n]$, $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}()$, $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
$(m, \sigma, f) \leftarrow \mathcal{A}^{\mathcal{SO}_{\vec{\mathsf{pk}}}}(\mathsf{pp}, \vec{\mathsf{pk}})$
return $(\mathsf{Verify}(\vec{\mathsf{pk}}, m, f, \sigma) = 1 \wedge (\sigma, \cdot, m, f)) \notin \mathcal{SO}_{\vec{\mathsf{pk}}}$

We are now ready to define the rest of the security properties. In the security games, we abuse notation and we denote with $\mathcal{SO}_{\vec{\mathsf{pk}}}, \mathcal{CO}$ both the oracles and the elements in the memory of the oracles.

Anonymity states that an adversary, given two signatures, cannot dectect the signer that signed them without knowing at least one of the two signers's secret key or having knowledge of a signature originating from one of the two signers.

Definition 5 (Anonymity). *An LRS scheme satisfies* anonymity *if for all probabilistic polynomial time adversaries* $\mathcal{A}$, $\Pr[\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A}}(\lambda) = 1] \leq 1/2 + \mathsf{negl}(\lambda)$.

Unforgeability states that an adversary that does not own any secret key of a ring, cannot produce a valid signature for that ring.

Definition 6 (Unforgeability). *An LRS scheme satisfies* linkability *if for all probabilistic polynomial time (PPT) adversaries* $\mathcal{A}$, $\Pr[\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

Linkability states that an adversary cannot create more unlinkable signatures than the number of secret keys they own.

Definition 7 (Linkability). *An LRS scheme satisfies* linkability *if for all PPT adversaries* $\mathcal{A}$, $\Pr[\mathsf{Exp}^{\mathsf{link}}_{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

Non-slanderability states that an adversary cannot create a signature that links to an honestly generated signature, without knowing the secret key of the signer. As in [7,15] we follow the 2-stage definition of Backes et al. [3] that grants the adversary access to all of the secret keys in the ring only after producing the slandering signature.

Game 3: Linkability game $\mathsf{Exp}^{\mathsf{link}}_{\mathcal{A}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
for $i \in [n]$, $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}()$, $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
$(\{m_i, \sigma_i\}_{i \in [n+1]}, f) \leftarrow \mathcal{A}^{\mathcal{CO}, \mathcal{SO}_{\vec{\mathsf{pk}}}}(\mathsf{pp}, \vec{\mathsf{pk}})$
return $((\forall i \in [n+1], \mathsf{Verify}(\vec{\mathsf{pk}}, m_i, f, \sigma_i) = 1) \wedge (\forall i, j \in [n+1], i \neq j,$
$\mathsf{Link}(\sigma_i, \sigma_j, f) = 0))$

Game 4: Non-slanderability game $\mathsf{Exp}^{\mathsf{non\text{-}sland}}_{\mathcal{A}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
for $i \in [n]$, $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KeyGen}()$, $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
$(m, f, \sigma) \leftarrow \mathcal{A}^{\mathcal{SO}_{\vec{\mathsf{pk}}}}(\mathsf{pp}, \vec{\mathsf{pk}})$
$(m', \sigma') \leftarrow \mathcal{A}^{\mathcal{SO}_{\vec{\mathsf{pk}}}, \mathcal{CO}}$
return $(\mathsf{Verify}(\vec{\mathsf{pk}}, m, f, \sigma) = 1 \wedge \mathsf{Verify}(\vec{\mathsf{pk}}, m', f, \sigma') = 1 \wedge \mathsf{Link}(\sigma, \sigma', f) =$
$1 \wedge (\sigma, \cdot, m, f) \notin \mathcal{SO}_{\vec{\mathsf{pk}}})$

Definition 8 (Non-slanderability). *An LRS scheme satisfies* non-slanderability *if for all PPT adversaries* $\mathcal{A}$, $\Pr[\mathsf{Exp}^{\mathsf{non\text{-}sland}}_{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

We note that throughout the paper, we present the Sign algorithm of the LRS schemes in its interactive version. The algorithm can be made non-interactive using the Fiat-Shamir transform [11], by replacing the verifier challenges with the output of a hash function on input the public statement and the prover's messages. The non-interactive signature verification algorithm can be derived from the interactive Sign algorithm by having the verifier re-construct the challenges using the statement and the data provided by the prover, and then perform the checks of the interactive verifier in Sign.

2.3 Discrete Log Groups

Prime order Groups Let $\mathbb{G}$ be a group of prime order q. Let G be a generator of $\mathbb{G}$. We denote this setup by $(q, \mathbb{G}, G)$. Due to space, we omit definitions for the DLP, and DDH problems for prime order groups.

Definition 9 (Discrete Log Relation). *Given* $\vec{G} \in \mathbb{G}^n$*, a non-trivial discrete log relation (DLR) over* $\mathbb{G}$ *is a vector* $\vec{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q$ *such that* $\vec{G}^{\vec{a}} = 1_{\mathbb{G}}$ *and* $\exists i : a_i \neq 0$.

Assumption 1 (DLR Assumption) *Given* $\vec{G} \leftarrow\!\!\$\ \mathbb{G}$*, no polynomial-time adversary can find a non-trivial Discrete Log Relation over* $\vec{G}$.

Pairing Groups Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a Type III pairing setup, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are of prime order q. Let G_1, G_2 be generators of $\mathbb{G}_1, \mathbb{G}_2$. We denote this setup by $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G_1, G_2)$.

Assumption 2 (SXDH [1]) *The* Symmetric eXternal Diffie-Hellman *assumption for* $(q, \mathbb{G}_1, \mathbb{G}_2, e, G_1, G_2)$ *as above, states that the Decisional Diffie Hellman assumption holds in both* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

SXDH implies the Double Pairing (DPair) assumption, defined below.

Assumption 3 (DPair) *For* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ *as above, given* $\vec{A} \leftarrow\$ \mathbb{G}_1^n$*, no non-uniform polynomial-time adversary can compute non-trivial* $\vec{B} \in \mathbb{G}_2^n$ *such that:* $e(\vec{A}, \vec{B}) = 1$*. Similarly, given* $\vec{B} \leftarrow\$ \mathbb{G}_2^n$*, no adversary can compute non-trivial* $\vec{A} \in \mathbb{G}_1$ *such that* $e(\vec{A}, \vec{B}) = 1$*.*

We write "DPair in $\mathbb{G}_1$" (resp. "DPair in $\mathbb{G}_2$") when the given vector is in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). We will use DPair in the following form, with a blinding term in $\mathbb{G}_T$.

Remark 1 (DPair with a blinding term). Under DPair in $\mathbb{G}_2$, given $\vec{B} \leftarrow\$ \mathbb{G}_2^n$ and $Q \leftarrow\$ \mathbb{G}_T$, no adversary can compute $\vec{A} \in \mathbb{G}_1^n$ and $r \in \mathbb{Z}_q$ with $(\vec{A}, r) \neq (\vec{1}, 0)$ such that $e(\vec{A}, \vec{B}) \cdot Q^r = 1$. Indeed, given a DPair instance $(\vec{B}, B_0) \leftarrow\$ \mathbb{G}_2^{n+1}$, a reduction samples $A_0 \leftarrow\$ \mathbb{G}_1$ and sets $Q = e(A_0, B_0)$, which is distributed as a uniform element of $\mathbb{G}_T$ up to negligible statistical distance. A solution $(\vec{A}, r)$ gives $e(\vec{A}, \vec{B}) \cdot e(A_0^r, B_0) = 1$, and $(\vec{A}, A_0^r)$ is non-trivial, since $A_0 \neq 1$ with overwhelming probability. The same holds with the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ swapped, under DPair in $\mathbb{G}_1$.

We also use the following assumption, the Generalised Pairing Inversion problem of [12].

Assumption 4 (GPI [12]) *For* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ *as above, given* $z \leftarrow\$ \mathbb{G}_T$*, no non-uniform polynomial-time adversary can compute* $A \in \mathbb{G}_1$ *and* $B \in \mathbb{G}_2$ *such that* $e(A, B) = z$*.*

2.4 Inner Product Arguments

Ring and linkable ring signature schemes use inner product arugments to reduce signature size and improve verification complexity [30,17,7,15].

Inner product arguments are arguments of knowledge. An argument of knowledge for a relation $\mathcal{R}$ consists of a tuple of interactive probabilistic polynomial time algorithms $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ that satisfies *perfect completeness* and *computational witness extended emulation* (CWEE). We provide the formal definition in Appendix A, together with the definitions of *perfect/statistical Honest Verifier Zero Knowledge* (HVZK) and *Witness Indistinguishability* (WI).

In the following sections we use the notation $\{x; w \mid \text{condition for } x, w\}$ to denote a relation that has an argument of knowledge in which the verifier's input is x and the prover's input is x, w.

Bulletproofs Bulletproofs [9] provide a logarithmic size and linear verifier time argument of knowledge for the relation

$$\{\vec{G}, \vec{H} \in \mathbb{G}^n, P \in \mathbb{G}, \hat{t} \in \mathbb{Z}_q; \vec{a}, \vec{b} \in \mathbb{Z}_q^n \mid P = \vec{G}^{\vec{a}} \cdot \vec{H}^{\vec{b}} \wedge \hat{t} = \langle \vec{a}, \vec{b} \rangle\} \tag{1}$$

Theorem 1 ([9]). *If Assumption 1 (DLR) holds in $\mathbb{G}$, w.r.t $\vec{G}$ and $\vec{H}$, the Bulletproof protocol is an argument of knowledge for the relation of Equation 1.*

Using Bulletproofs, we can construct a logarithmic size argument of knowledge for range checks, i.e. for the relation

$$\{G, Q \in \mathbb{G}, \mathsf{cm} \in \mathbb{G}, n \in \mathbb{N}; v, r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = G^v Q^{r_{\mathsf{cm}}} \wedge v \in [0, 2^n - 1]\} \quad (2)$$

Theorem 2 ([9]). *If the Discrete Log Relation assumption holds in $\mathbb{G}$, the protocol presented in [9] is a perfect HVZK argument of knowledge for the range relation of Equation 2.*

Dory Dory [18] is a logarithmic size and logarithmic verification time argument of knowledge for the relation

$$\begin{aligned}\{\vec{\Lambda} \in \mathbb{G}_1^n, \vec{\Gamma} \in \mathbb{G}_2^n, Q \in \mathbb{G}_T, D_0, D_1, D_2 \in \mathbb{G}_T; \vec{\Omega} \in \mathbb{G}_1^n, \vec{\Theta} \in \mathbb{G}_2^n, r_0, r_1, r_2 \in \mathbb{Z}_q \mid \\ e(\vec{\Omega}, \vec{\Theta}) \cdot Q^{r_0} = D_0, \quad e(\vec{\Omega}, \vec{\Gamma}) \cdot Q^{r_1} = D_1, \quad e(\vec{\Lambda}, \vec{\Theta}) \cdot Q^{r_2} = D_2\}\end{aligned} \quad (3)$$

Theorem 3 ([18]). *If $\vec{\Lambda} \leftarrow\!\!\$\ \mathbb{G}_1^n, \vec{\Gamma} \leftarrow\!\!\$\ \mathbb{G}_2^n$ and the SXDH assumption holds for $(q, \mathbb{G}_1, \mathbb{G}_2, e, G_1, G_2)$, then Dory is a statistical HVZK argument of knowledge for the relation of Equation 3.*

2.5 Omniring

Omniring [17] is a discrete logarithm-based linkable ring signature (LRS) scheme that uses Bulletproofs to achieve logarithmic signature size. For consistency, we use the description of Omniring provided in `LLRing`[15]. This will be useful when focusing on the differences between the two systems.

The setup of Omniring consists of a group $\mathbb{G}$ of prime order q for which the DLOG problem is assumed to be hard, a hash function $\mathsf{H}_G : \{0,1\}^* \to \mathbb{G}$, and for up to n users generators $(\vec{R}, \vec{H}, K, F, P, Q) \in \mathbb{G}^{2n+4}$. The KeyGen algorithm selects $\mathsf{sk} \leftarrow\!\!\$\ \mathbb{Z}_q$ and sets $\mathsf{pk} = P^{\mathsf{sk}}$.

The Sign algorithm of Omniring consists of two steps. Assume a prover who knows a secret key sk corresponding to a public key $\mathsf{pk}_{i^*} = P^{\mathsf{sk}} \in \mathbb{G}$ belonging to a ring of public keys $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n) \in \mathbb{G}^n$, with $i^* \in [n]$. To produce a signature, the prover first constructs a commitment $\mathsf{cm} = P^{\mathsf{sk}} Q^{r_{\mathsf{cm}}} = \mathsf{pk}_{i^*} \cdot Q^{r_{\mathsf{cm}}}$, where $r_{\mathsf{cm}} \leftarrow\!\!\$\ \mathbb{Z}_q$. Then, the prover proves in zero knowledge that they know an index $i^* \in [n]$ and value $r_{\mathsf{cm}} \in \mathbb{Z}_q$ such that $\mathsf{cm} = \mathsf{pk}_{i^*} \cdot Q^{r_{\mathsf{cm}}}$, meaning they engage in a zero knowledge argument of knowledge for the relation

$$\{Q, \mathsf{cm} \in \mathbb{G}, \vec{\mathsf{pk}} \in \mathbb{G}^n; i^* \in [n], r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = \mathsf{pk}_{i^*} \cdot Q^{r_{\mathsf{cm}}}\} \quad (4)$$

We will call the relation of Equation 4 the *ring membership* relation.

Then, the prover constructs their linking tag tag for an event f by setting $\mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}$. Finally, the prover proves in zero knowledge that they know $\mathsf{sk}, r_{\mathsf{cm}}$ such that $\mathsf{cm} = P^{\mathsf{sk}}Q^{r_{\mathsf{cm}}}$ and $\mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}$ by engaging in a zero knowledge argument of knowledge for the relation

$$\{P, Q, \mathsf{cm}, \mathsf{tag} \in \mathbb{G}, f \in \{0,1\}^*; \mathsf{sk}, r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = P^{\mathsf{sk}} \cdot Q^{r_{\mathsf{cm}}}, \mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}\} \tag{5}$$

We will call the relation of Equation 5 the *tag* relation.

Finally, the Link algorithm given two signatures with linking tags $\mathsf{tag}_1, \mathsf{tag}_2$, checks whether $\mathsf{tag}_1 = \mathsf{tag}_2$.

Ring membership proof For the zero knowledge argument of knowledge for the ring membership relation, the prover, having set $\mathsf{cm} = \mathsf{pk}_{i^*} \cdot Q^{r_{\mathsf{cm}}}$, does the following.

1. The prover constructs $\vec{c}$ such that $c_i = \begin{cases} 1, & \text{if } i = i^* \\ 0, & \text{otherwise} \end{cases}$. Notice that $\mathsf{cm} = \vec{\mathsf{pk}}^{\vec{c}} \cdot Q^{r_{\mathsf{cm}}}$, and the prover should show that $\vec{c}$ is binary and sums to 1. The prover sets $\vec{c}' = \vec{c} - \vec{1}$, which will be used to help prove that $\vec{c}$ is binary.
2. The prover commits to $\vec{c}, \vec{c}'$ using generators[6] $\vec{R}, \vec{H}, Q$: $\hat{A} = \vec{R}^{\vec{c}}\vec{H}^{\vec{c}'}Q^{\hat{r}_A}$ where $\hat{r}_A \leftarrow\!\!\$\ \mathbb{Z}_q$, and sends $\hat{A}$ to the verifier.
3. The verifier sends $w \leftarrow\!\!\$\ \mathbb{Z}_q$, and both the prover and the verifier construct generators $\vec{G} = \vec{\mathsf{pk}}^{w} \circ \vec{R} = (\mathsf{pk}_1^w \cdot R_1, \ldots, \mathsf{pk}_n^w \cdot R_n)$.
4. The prover calculates $A = \mathsf{cm}^w\hat{A} = \vec{G}^{\vec{c}}\vec{H}^{\vec{c}'}Q^{r_A}$, where $r_A = w \cdot r_{\mathsf{cm}} + \hat{r}_A$, and sends A to the verifier. The prover also commits to blinding vectors $\vec{s}_1, \vec{s}_2 \leftarrow\!\!\$\ \mathbb{Z}_q^n$: $S = \vec{G}^{\vec{s}_1}\vec{H}^{\vec{s}_2}Q^{r_S}$ using $r_S \leftarrow\!\!\$\ \mathbb{Z}_q$ and sends S to the verifier.
5. Then, the prover proves in zero knowledge that they know vectors $\vec{c}, \vec{c}'$ such that
$$\langle \vec{c}, \vec{1} \rangle = 1 \text{ and } \vec{c} - \vec{1} - \vec{c}' = \vec{0} \text{ and } \vec{c} \circ \vec{c}' = \vec{0} \tag{6}$$
and such that $A = \vec{G}^{\vec{c}}\vec{H}^{\vec{c}'}Q^{r_A}$. Equation 6 is equivalent to $\vec{c}$ being binary and summing to 1. The proof is constructed using a modification of the range proof protocol of Bulletproofs.
6. Finally, the verifier also checks that $A = \mathsf{cm}^w\hat{A}$.

The random value w given by the verifier on step 3 is to ensure that cm contains only elements of $\vec{\mathsf{pk}}$ and Q, and $\hat{A}$ only elements of $\vec{R}$ and Q. Otherwise, the prover could for example set $\mathsf{cm} = \mathsf{pk}_1\mathsf{pk}_2Q^{r_{\mathsf{cm}}}$ and $\hat{A} = R_2 \cdot (\mathsf{pk}_1)^{-1}Q^{\hat{r}_A}$ and the proof would still pass verification.

To prove that $\vec{c}, \vec{c}'$ satisfy Equation 6 and that $A = \vec{G}^{\vec{c}}\vec{H}^{\vec{c}'}Q^{r_A}$ (step 5.), the prover, similarly to the range proof protocol of Bulletproofs, first receives

[6] Note that the prover does not know the DLOG relations between elements of $\vec{R}, \vec{H}, P, Q$.

$y, z \leftarrow\!\!\$\ \mathbb{Z}_q$ from the verifier, sets $\vec{y}^n = (1, y, y^2, \ldots, y^{n-1}) \in \mathbb{Z}_q^n$, and constructs vectors of polynomials

$$\vec{l}(X) = \vec{c} + X \cdot \vec{s}_1 - z \cdot \vec{1}$$
$$\vec{r}(X) = \vec{y}^n \circ (\vec{c}' + X \cdot \vec{s}_2 + z \cdot \vec{1}) + z^2 \cdot \vec{1}.$$

These polynomials are constructed such that if the constant term of the polynomial $\hat{t}(X) = \langle l(X), r(X) \rangle$ is a specific value $\delta(y, z)$, then $\vec{c}, \vec{c}'$ satisfy Equation 6. The prover needs to prove that the constant term of $\hat{t}(X)$ is $\delta(y, z)$ and that $\vec{l}(X), \vec{r}(X)$ are correct given the initial commitments to $\vec{c}, \vec{c}', \vec{s}_1, \vec{s}_2$.

Let $\hat{t}(X) = \delta(y, z) + t_1 \cdot X + t_2 \cdot X^2$. The prover first constructs commitments $T_1 = F^{t_1} \cdot Q^{\tau_1}, T_2 = F^{t_2} \cdot Q^{\tau_2}$, where $\tau_1, \tau_2 \leftarrow\!\!\$\ \mathbb{Z}_q$ and then evaluates $\vec{l}(X), \vec{r}(X)$ to a point $x \leftarrow\!\!\$\ \mathbb{Z}_q$ sent by the verifier, setting $\vec{l} = \vec{l}(x), \vec{r} = \vec{r}(x)$. The prover sends $\hat{t}(x) = \hat{t} = \langle \vec{l}, \vec{r} \rangle$ together with a response τ_x for the base-Q of the commitments, and the verifier can check that

$$F^{\hat{t}} \cdot Q^{\tau_x} = F^{\delta(y,z)} \cdot T_1^x \cdot T_2^{x^2},$$

thus checking that the constant term of $\hat{t}(X)$ is indeed the value $\delta(y, z)$.

Regarding the verification of the construction of $\vec{l}, \vec{r}$, the prover could send $\vec{l} = \vec{l}(x), \vec{r} = \vec{r}(x)$ and a response r_W for the base-Q parts of the commitments, and the verifier can check that

$$\vec{G}^{\vec{l}} (\vec{H}^{\circ \vec{y}^{-n}})^{\vec{r}} = A \cdot S^x \cdot \vec{G}^{-z \cdot \vec{1}} \cdot (H^{\circ \vec{y}^{-n}})^{z \cdot \vec{y}^n + z^2 \cdot \vec{1}},$$

where $\vec{y}^{-n} = (1, y^{-1}, y^{-2}, \ldots, y^{-n+1})$. However, sending $\vec{l}, \vec{r}$ would result in a signature of linear size. Instead, the prover sends the values $W = \vec{G}^{\vec{l}} (\vec{H}^{\circ \vec{y}^{-n}})^{\vec{r}}$ and $\hat{t} = \hat{t}(x)$ together with a Bulletproof for the correctness of $W, \hat{t}$. Finally, the verifier checks the correct construction of $\vec{l}, \vec{r}$ given the initial commitments, plus the fact that the constant term of $\hat{t}(X)$ is $\delta(y, z)$, by checking the equations

$$\begin{cases} F^{\hat{t}} \cdot Q^{\tau_x} = F^{\delta(y,z)} \cdot T_1^x \cdot T_2^{x^2} \\ W \cdot Q^{r_W} = A \cdot S^x \cdot \vec{G}^{-z \cdot \vec{1}} \cdot (H^{\circ \vec{y}^{-n}})^{z \cdot \vec{y}^n + z^2 \cdot \vec{1}} \end{cases}$$

Tag proof Given a well-formed commitment $\mathsf{cm} = P^{\mathsf{sk}} \cdot Q^{r_{\mathsf{cm}}}$ and $\mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}$, the prover uses the tag protocol of Fig. 1 to prove the tag relation of Eq. 5.

Theorem 4 ([7]). *The protocol of Figure 1 is a HVZK argument of knowledge for the relation of Figure 1 under the DLR assumption.*

3 Unlinkability attacks on the LLRing Schemes

3.1 On the unlinkability of LLRing-DL

Description of LLRing-DL The `LLRing-DL` scheme applies two modifications to the ring membership proof of Omniring in order to improve efficiency.

Protocol for the relation

$$\{P, Q, \mathsf{cm}, \mathsf{tag} \in \mathbb{G}, f \in \{0,1\}^*; \mathsf{sk}, r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = P^{\mathsf{sk}} \cdot Q^{r_{\mathsf{cm}}}, \mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}\}$$

- $\mathcal{P} \to \mathcal{V}$: $A = \mathsf{H}_G(f)^r, B = P^r \cdot Q^s$ for $r, s \leftarrow\!\!\$\, \mathbb{Z}_q$
- $\mathcal{P} \leftarrow \mathcal{V}$: $x \leftarrow\!\!\$\, \mathbb{Z}_q$
- $\mathcal{P} \to \mathcal{V}$: $z = r + x \cdot \mathsf{sk}, w = s + x \cdot r_{\mathsf{cm}}$
- $\mathcal{V}$: check that $A \cdot \mathsf{tag}^x = \mathsf{H}_G(f)^z$ and $B \cdot \mathsf{cm}^x = P^z \cdot Q^w$.

Fig. 1. Protocol to prove correctness of a linking tag.

First, it reduces the number of group exponentations performed by the verifier by replacing Bulletproofs with Segregated Bulletproofs, which is a modification of Bulletproofs introduced in the paper. Intuitively a segregated Bulletproof replaces a single proof of size $2n$ with a pair of related proofs of size n, run concurrently. As the dominant computation cost for the prover is the folding of the basis vectors, starting out with vectors of half the size greatly increases speed. Thus, the verification time is improved, at the cost of doubling the size.

When using Segregated Bulletproofs, instead of committing to $\vec{c}, \vec{c}'$ in one commitment $A = \vec{G}^{\vec{c}}\vec{H}^{\vec{c}'}Q^{r_A}$, for $\vec{G} \neq \vec{H}$, $\vec{c}$ and $\vec{c}'$ are committed seperately, by setting $\vec{G} = \vec{H}$ and $A = \vec{G}^{\vec{c}}Q^{r_A}$, $B = \vec{H}^{\vec{c}'}Q^{r_B} = \vec{G}^{\vec{c}'}Q^{r_B}$. The blinding vectors for $\vec{c}, \vec{c}'$ are also seperately committed, by setting $S_1 = \vec{G}^{\vec{s_1}}Q^{r_{S_1}}, S_2 = \vec{H}^{\vec{s_2}}Q^{r_{S_2}}$. Thus the verification equation for the correctness of $\vec{l}, \vec{r}$ is split into two equations

$$\begin{cases} \vec{G}^{\vec{l}} \cdot Q^{r_{W_1}} = A \cdot S_1^x \cdot \vec{G}^{-z \cdot \vec{1}} \\ (\vec{H}^{\circ \vec{y}^{-n}})^{\vec{r}} \cdot Q^{r_{W_2}} = B \cdot S_2^x \cdot (\vec{H}^{\circ \vec{y}^{-n}})^{z \cdot \vec{y}^n + z^2 \cdot \vec{1}} \end{cases}$$

Second, to allow for pre-proccessing, and further reduce the number of group exponentiations of the prover and the verifier, the vector $\vec{R} = (R_1, \ldots, R_n)$ of step 2 of Omniring is replaced with $\vec{R} = (R^{\mathsf{H}(\mathsf{pk}_1)}, \ldots, R^{\mathsf{H}(\mathsf{pk}_n)})$ for a hash function H. This avoids the verifier having to send the value w. To ensure that cm contains only elements of $\vec{\mathsf{pk}}$ and $\hat{A}$ only elements of $\vec{R}$, the prover also provides a Schnorr proof that they know the representation of $\hat{A}$ in base R, Q.

This modification redefines $\vec{G}$ as $\vec{G} = \vec{\mathsf{pk}} \circ \vec{R} = (\mathsf{pk}_1 \cdot R^{h_1}, \ldots, \mathsf{pk}_n \cdot R^{h_n})$ (where $h_i = \mathsf{H}(\mathsf{pk}_i)$). This causes $\vec{G}$ to be independent of the signature so that it may be computed during a pre-proccessing phase.

Finally, the vector $\vec{H}$ is set as $\vec{H} = \vec{G}$, so the setup of `LLRing-DL` is

$$(R, K, F, P, Q) \in \mathbb{G}^5$$

We will show that the second of the two `LLRing-DL` modifications to the ring membership proof leads to an unlinkability attack. The reason is that the security of the Bulletproofs range proof (and of the Omniring protocol) is based on the fact that it is hard to find non-trivial DLOG relations between the generators. However, the redefined basis vector $\vec{G}$ is highly structured, so that finding discrete log relations is in fact tractable.

Warmup: Finding Non-Trivial DLOG Relations Observe that by using $\vec{R} = (R^{h_1}, \dots R^{h_n})$ and $\vec{G} = \vec{R} \circ \vec{\mathsf{pk}}$, if a prover knows $m > 1$ secret keys, they can find non-trivial DLOG relations for the vector of generators $\vec{G}$: They just have to find $\vec{a} \in \mathbb{Z}_q$ such that

$$\prod_{i=1}^{n} G_i^{a_i} = 1 \Leftrightarrow \prod_{i=1}^{n} (\mathsf{pk}_i \cdot R^{h_i})^{a_i} = 1 \Leftrightarrow \prod_{i=1}^{n} \mathsf{pk}_i^{a_i} \cdot \prod_{i=1}^{n} R^{h_i \cdot a_i} = 1$$

Assume w.l.o.g. that the prover knows the first m secret keys. The prover can set $a_i = 0$ for $i \in \{m+1, \dots, n\}$ and solve the system of equations

$$\begin{cases} \sum_{i=1}^{m} \mathsf{sk}_i \cdot a_i = 0 \\ \sum_{i=1}^{m} h_i \cdot a_i = 0 \end{cases}$$

This gives us a system of m unknowns and 2 equations with random coefficients, so the system has more than one solutions.

The above analysis contradicts the assumption that finding non-trivial DLOG relations between generators $\vec{G}$ is hard. That assumption is required for the soundness of the Bulletproofs range proof used in the protocol. Furthermore, we are able to describe a concrete attack based on solving the two equations.

Implementing an unlinkability attack on `LLRing-DL` Assume a malicious prover $\mathcal{P}^*$ that w.l.o.g. knows the first m secret keys of the ring $\vec{\mathsf{pk}}$. The prover $\mathcal{P}^*$ picks $a_i \leftarrow\!\!\$\ \mathbb{Z}_q$ for $i \in [m]$, sets $a_i = 0$ for $i \in \{m+1, \dots, n\}$ and sets $\mathsf{cm} = \vec{\mathsf{pk}}^{\vec{a}} Q^{r_{\mathsf{cm}}}$. If $\mathcal{P}^*$ manages to create a verifying proof for cm for the ring membership protocol, they can then set $\mathsf{tag} = \mathsf{H}_G(f)^{\sum_{i \in [m]} \mathsf{sk}_i a_i}$ and create a valid tag proof using $y = \sum_{i \in [m]} \mathsf{sk}_i a_i, r_{\mathsf{cm}}$ as witness. That is, $\mathsf{cm} = P^y Q^{r_{\mathsf{cm}}}$ and $\mathsf{tag} = \mathsf{H}_G(f)^y$, but $y \neq \mathsf{sk}_j$ for any sk_j corresponding to a pk_j of the ring. The vector $\vec{a}$ can be chosen at random, so $\mathcal{P}^*$ can create an unlinkable signature for any linear combination of secret keys they own.

We now show how $\mathcal{P}^*$ can create a verifying proof for $\mathsf{cm} = \vec{\mathsf{pk}}^{\vec{a}} Q^{r_{\mathsf{cm}}} = P^{\sum_{i \in [m]} a_i \mathsf{sk}_i} Q^{r_{\mathsf{cm}}}$ for the ring membership protocol.

1. First, $\mathcal{P}^*$ sets $\hat{A} = R^{\sum_{i \in [m]} a_i h_i} Q^{\hat{r}_A}$ where $\hat{r}_A \leftarrow\!\!\$\ \mathbb{Z}_q$ and proceeds to run the Schnorr protocol to prove knowledge of the DLOG of A in base R, Q. Note that $\mathcal{P}^*$ is able to provide a verifying proof since they indeed know the DLOG of $\hat{A}$ base R, Q.
2. Both $\mathcal{P}^*$ and the verifier continue the protocol as normal with the only difference being that $\mathcal{P}^*$ sets $A = \vec{G}^{\vec{a}} Q^{r_A}$. For the commitment B, $\mathcal{P}^*$ picks an arbitrary index $i^* \in [m]$, constructs the vectors $\vec{c}$ and $\vec{c}' = \vec{c} - \vec{1}$ honestly for i^*, and commits to $B = \vec{G}^{\vec{c}'} Q^{r_B}$. Finally, $\mathcal{P}^*$ creates the commitments S_1, S_2 to the blinding factors as in the protocol.
3. After getting the randomness y, z from the verifier, $\mathcal{P}^*$ constructs the polynomial coefficients t_1 and t_2 using $\vec{c}, \vec{c}'$, and commits to t_1, t_2 as in the `LLRing-DL` protocol. $\mathcal{P}^*$ also creates $\vec{l}, \vec{r}$ using $\vec{c}, \vec{c}'$ as in the protocol and sets $\hat{t} = \langle \vec{l}, \vec{r} \rangle$.

4. As a simplification, suppose that in the final step of LLRing-DL we have Bulletproofs finish early by having the prover send the full left and right vectors $\vec{l}, \vec{r}$ in the open, and the verifier calculates $\hat{t} = \langle \vec{l}, \vec{r} \rangle$ themselves. This change can only improve soundness, meaning the attack also works in the unmodified protocol. The prover $\mathcal{P}^*$ cannot send $\vec{l}, \vec{r}$, as they will not pass verification, since the values $A, \hat{A}$ commit to $\vec{a}$ instead of $\vec{c}$. Instead $\mathcal{P}^*$ can compute a different $\vec{l'}$ vector that *will* pass verification. As a reminder, the verification equations are the following.

$$\begin{cases} A = \mathsf{cm} \cdot \hat{A} \\ F^{\hat{t}} \cdot Q^{\tau_x} = F^{\delta(y,z)} \cdot T_1^x \cdot T_2^{x^2} \\ \vec{G}^{\vec{l}} \cdot Q^{r_{W_1}} = A \cdot S_1^x \cdot \vec{G}^{-z \cdot \vec{1}} \\ (\vec{H}^{\circ \vec{y}^{-n}})^{\vec{r}} \cdot Q^{r_{W_2}} = B \cdot S_2^x \cdot (\vec{H}^{\vec{y}^{-n}})^{z \cdot \vec{y}^n + z^2 \cdot \vec{1}} \end{cases}$$

The first equation is satisfied from the construction of $A, \mathsf{cm}, \hat{A}$, and the the fourth equation is satisfied from the construction of B. In order for the second equation to be satisfied, it should hold that $\langle \vec{l'}, \vec{r} \rangle = \hat{t}$, and in order for the third equation to be satisfied, it should hold that $\vec{G}^{\vec{l'}} = \vec{G}^{\vec{a} + x \cdot \vec{s_1} - z \cdot \vec{1}}$.
For $i \in \{m+1, \ldots, n\}$, $\mathcal{P}^*$ can set $l'_i = l_i = x \cdot s_{1i} - z$. For $i \in [m]$, we have that

$$\langle \vec{l'}, \vec{r} \rangle = \hat{t} \Leftrightarrow \langle \vec{l'}, \vec{r} \rangle = \langle \vec{l}, \vec{r} \rangle \Leftrightarrow \sum_{i \in [m]} l'_i \cdot r_i = \hat{t} - \sum_{i=m+1}^{n} l_i \cdot r_i$$

and

$$\begin{aligned} \vec{G}^{\vec{l'}} = \vec{G}^{\vec{a} + x \cdot \vec{s_1} - z \cdot \vec{1}} &\Leftrightarrow \prod_{i \in [m]} G_i^{l'_i} \prod_{i=m+1}^{n} G_i^{l'_i} = \prod_{i \in [m]} G_i^{a_i + x \cdot s_{1i} - z} \prod_{i=m+1}^{n} G_i^{x \cdot s_{1i} - z} \\ &\Leftrightarrow \prod_{i \in [m]} G_i^{l'_i} = \prod_{i \in [m]} G_i^{a_i + x \cdot s_{1i} - z} \\ &\Leftrightarrow \prod_{i \in [m]} \mathsf{pk}_i^{l'_i} \prod_{i \in [m]} (R^{h_i})^{l'_i} = \prod_{i \in [m]} \mathsf{pk}_i^{a_i + x \cdot s_{1i} - z} \prod_{i \in [m]} (R^{h_i})^{a_i + x \cdot s_{1i} - z} \\ &\Leftrightarrow P^{\sum_{i \in [m]} \mathsf{sk}_i \cdot l'_i} R^{\sum_{i \in [m]} h_i \cdot l'_i} = P^{\sum_{i \in [m]} \mathsf{sk}_i \cdot (a_i + x \cdot s_{1i} - z)} R^{\sum_{i \in [m]} h_i \cdot (a_i + x \cdot s_{1i} - z)} \end{aligned}$$

So $\mathcal{P}^*$ needs to find $l'_i, i \in [m]$ such that

$$\begin{cases} \sum_{i \in [m]} l'_i \cdot r_i = \hat{t} - \sum_{i=m+1}^{n} l_i \cdot r_i \\ \sum_{i \in [m]} l'_i \cdot \mathsf{sk}_i = \sum_{i \in [m]} \mathsf{sk}_i \cdot (a_i + x \cdot s_{1i} - z) \\ \sum_{i \in [m]} l'_i \cdot h_i = \sum_{i \in [m]} h_i \cdot (a_i + x \cdot s_{1i} - z) \end{cases}$$

We conclude that $\mathcal{P}^*$ has to find a solution to a system of three equations with random coefficients and m unknowns. If $m > 3$, the attack succeeds.

3.2 Restoring the linkability of LLRing-DL

In Appendix B, we demonstrate how to restore the linkability of `LLRing-DL`, by modifying the base selection so that vector $\vec{R}$ consists of random group elements $(R_1, \ldots, R_n)$. We maintain the use of segregated Bulletproofs with appropriate modifications but note that the speed / size tradeoff is impacted by the structure of $\vec{R}$, resulting in a protocol with broadly similar performance to Omniring.

3.3 On the unlinkability of LLRing-P

Description of LLRing-P The `LLRing-P` scheme uses Dory to achieve logarithmic verification complexity. Similarly to Omniring and `LLRing-DL`, the Sign algorithm consists of protocols for a ring membership relation and a tag relation.

The setup of `LLRing-P` consists of a type III pairing setup $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, generators $(P, R, L') \leftarrow\!\$\ \mathbb{G}_1^3, L \leftarrow\!\$\ \mathbb{G}_2, \vec{H} \leftarrow\!\$\ \mathbb{G}_2^n, Q \leftarrow\!\$\ \mathbb{G}_T$ and a hash function $\mathsf{H}_G : \{0,1\}^* \to \mathbb{G}_1$. Denote $\vec{L'} = (L', \ldots, L') \in \mathbb{G}_1^n$, $\vec{L} = (L, \ldots, L) \in \mathbb{G}_2^n$. Public keys are base P, meaning $\mathsf{pk} = P^{\mathsf{sk}} \in \mathbb{G}_1$. Similarly to `LLRing-DL`, for a ring of public keys $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n) \in \mathbb{G}_1^n$ we set $\vec{R} = (R^{\mathsf{H}(\mathsf{pk}_1)}, \ldots, R^{\mathsf{H}(\mathsf{pk}_n)}) = (R^{h_1}, \ldots, R^{h_n}) \in \mathbb{G}_1^n$ and $\vec{G} = \vec{\mathsf{pk}} \circ \vec{R} \in \mathbb{G}_1^n$.

The Sign algorithm works as follows. A prover with index i^* in a ring of public keys $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n) \in \mathbb{G}_1^n$ first commits to their public key pk_{i^*} using an AFGHO commitment [1] by setting $\mathsf{cm} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm}}}$, where $r_{\mathsf{cm}} \leftarrow\!\$\ \mathbb{Z}_q$. Then, the prover engages in an argument of knowledge for the ring membership relation.

$$\{L \in \mathbb{G}_2, Q, \mathsf{cm} \in \mathbb{G}_T, \vec{\mathsf{pk}} \in \mathbb{G}_1^n; i^* \in [n], r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm}}}\} \quad (7)$$

Then, the prover constructs the linking tag $\mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}$ and proves in zero knowledge that cm and tag share the same secret key, by using a zero knowledge argument of knowledge for the tag relation

$$\begin{aligned}\{P \in \mathbb{G}_1, Q, \mathsf{cm} \in \mathbb{G}_T, \mathsf{tag} \in \mathbb{G}_1, f \in \{0,1\}^*; \mathsf{sk}, r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \\ \mathsf{cm} = e(P^{\mathsf{sk}}, L) \cdot Q^{r_{\mathsf{cm}}}, \mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}\}\end{aligned} \quad (8)$$

The Link algorithm works as in Omniring: given two signatures with linking tags $\mathsf{tag}_1, \mathsf{tag}_2$, checks whether $\mathsf{tag}_1 = \mathsf{tag}_2$.

Ring membership proof The ring membership protocol for $\mathsf{cm} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm}}}$ works as follows.

The prover first constructs $\vec{c}$ such that $c_i = \begin{cases} 1, & \text{if } i = i^* \\ 0, & \text{otherwise} \end{cases}$ and commits to $\vec{c}$ using AFGHO commitments, as follows

$$\begin{aligned}\hat{A} &= e(R, L)^{h_{i^*}} Q^{\hat{r}_A} = e(\vec{R}, \vec{L})^{\vec{c}} \cdot Q^{\hat{r}_A}, \hat{r}_A \leftarrow\!\$\ \mathbb{Z}_q \\ B &= e(\vec{G}, \vec{H})^{\vec{c}} \cdot Q^{r_B}, r_B \leftarrow\!\$\ \mathbb{Z}_q.\end{aligned}$$

Then, the prover proves knowledge of the discrete logarithm of $\hat{A}$ base $e(R, L), Q$, similarly to `LLRing-DL`.

Then the prover sets $A = \mathsf{cm} \cdot \hat{A}$ and $r_A = r_{\mathsf{cm}} + \hat{r}_A$, and needs to prove knowledge of the unit basis vector $\vec{c}$ and $r_A, r_B \in \mathbb{Z}_q$ for which

$$A = e(\vec{G}, \vec{L})^{\vec{c}} \cdot Q^{r_A} \text{ and } B = e(\vec{G}, \vec{H})^{\vec{c}} \cdot Q^{r_B} \tag{9}$$

To prove that $\vec{c}$ is a unit basis vector, the prover equivalently proves that the following holds for $\vec{c}$

$$\langle \vec{c}, \vec{1} \rangle = 1 \text{ and } \vec{c} \circ (\vec{c} - \vec{1}) = \vec{0} \tag{10}$$

To prove the relations of Equation 9 and Equation 10, the prover uses Dory to prove three equivalent sets of relations.

Setting $\vec{\Omega}' = \vec{L}', \vec{\Theta}' = \vec{L}^{\circ \vec{c}}, r'_0 = r'_1 = 0, r'_2 = r_A$, the prover proves

$$\begin{aligned} \{\vec{G} \in \mathbb{G}_1^n, \vec{H} \in \mathbb{G}_2^n, Q, e(L', L), e(\vec{L}', \vec{H}), A \in \mathbb{G}_T; \vec{\Omega}' \in \mathbb{G}_1^n, \vec{\Theta}' \in \mathbb{G}_2^n, r'_0, r'_1, r'_2 \in \mathbb{Z}_q \mid \\ e(\vec{\Omega}', \vec{\Theta}') \cdot Q^{r'_0} = e(L', L), \\ e(\vec{\Omega}', \vec{H}) \cdot Q^{r'_1} = e(\vec{L}', \vec{H}), \\ e(\vec{G}, \vec{\Theta}') \cdot Q^{r'_2} = A\} \end{aligned} \tag{11}$$

which is used to prove that $\langle \vec{c}, \vec{1} \rangle = 1$ and that $\vec{c}$ is committed in A.

Setting $\vec{\Omega}'' = \vec{G}^{\circ \vec{c}}, \vec{\Theta}'' = \vec{H}^{\circ \vec{c}}, r''_0 = r''_1 = r''_2 = r_B$, the prover proves

$$\begin{aligned} \{\vec{G} \in \mathbb{G}_1^n, \vec{H} \in \mathbb{G}_2^n, Q, B \in \mathbb{G}_T; \vec{\Omega}'' \in \mathbb{G}_1^n, \vec{\Theta}'' \in \mathbb{G}_2^n, r''_0, r''_1, r''_2 \in \mathbb{Z}_q \mid \\ e(\vec{\Omega}'', \vec{\Theta}'') \cdot Q^{r''_0} = B, \\ e(\vec{\Omega}'', \vec{H}) \cdot Q^{r''_1} = B, \\ e(\vec{G}, \vec{\Theta}'') \cdot Q^{r''_2} = B\} \end{aligned} \tag{12}$$

which is used to prove that $\vec{c} \circ (\vec{c} - \vec{1}) = \vec{0}$ and that $\vec{c}$ is committed in B.

Finally, setting $\vec{\Omega}''' = \vec{G}^{\circ \vec{c}}, \vec{\Theta}''' = \vec{L}, r'''_0 = r_A, r'''_1 = r_B, r'''_2 = 0$, the prover proves

$$\begin{aligned} \{\vec{G} \in \mathbb{G}_1^n, \vec{H} \in \mathbb{G}_2^n, Q, A, B, e(\vec{G}, \vec{L}) \in \mathbb{G}_T; \vec{\Omega}''' \in \mathbb{G}_1^n, \vec{\Theta}''' \in \mathbb{G}_2^n, r'''_0, r'''_1, r'''_2 \in \mathbb{Z}_q \mid \\ e(\vec{\Omega}''', \vec{\Theta}''') \cdot Q^{r'''_0} = A, \\ e(\vec{\Omega}''', \vec{H}) \cdot Q^{r'''_1} = B, \\ e(\vec{G}, \vec{\Theta}''') \cdot Q^{r'''_2} = e(\vec{G}, \vec{L})\} \end{aligned} \tag{13}$$

which ensures that the same $\vec{c}$ is committed in both A and B.

Tag proof Given $\mathsf{cm} = e(P, L)^{\mathsf{sk}} \cdot Q^{r_{\mathsf{cm}}}$, the prover sets $\mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}$ and then uses the protocol of Figure 2.

Protocol for the relation

$$\{P \in \mathbb{G}_1, Q, \mathsf{cm} \in \mathbb{G}_T, \mathsf{tag} \in \mathbb{G}_1, f \in \{0,1\}^*; \mathsf{sk}, r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = e(P^{\mathsf{sk}}, L) \cdot Q^{r_{\mathsf{cm}}}, \mathsf{tag} = \mathsf{H}_G(f)^{\mathsf{sk}}\}$$

$\mathcal{P}$: $r, s \leftarrow\!\$\, \mathbb{Z}_q$
$\mathcal{P} \rightarrow \mathcal{V}$: $A = \mathsf{H}_G(f)^r, B = e(P^r, L) \cdot Q^s$
$\mathcal{P} \leftarrow \mathcal{V}$: $x \leftarrow\!\$\, \mathbb{Z}_q$
$\mathcal{P} \rightarrow \mathcal{V}$: $z = r + x \cdot \mathsf{sk}, w = s + x \cdot r_{\mathsf{cm}}$
$\mathcal{V}$: check that $A \cdot \mathsf{tag}^x = \mathsf{H}_G(f)^z$ and $B \cdot \mathsf{cm}^x = e(P^z, L) \cdot Q^w$

Fig. 2. Protocol to prove correctness of a linking tag.

Implementing an unlinkability attack on LLRing-P This attack exploits the same weakness as our attack on `LLRing-DL`, namely the weak structure of the $\vec{G}$ basis vector. As $G_i := R^{h_i} \cdot P^{\mathsf{sk}_i}$, knowing multiple keys allows a malicious prover to "balance" the corresponding G_i elements with non trivial factors so that they cancel out. Similar to the previous attack, the malicious prover will commit to an arbitrary linear combination of keys in A and to a bitstring for B, which does not have to be of weight 1. Intuitively this is because the malicious prover is able to leverage the structure of $\vec{G}$ on pairing checks that involve $\vec{G}$ and $\vec{L}' := (L' \ldots, L')$ (as all elements of $\vec{L}'$ share the same log) but not those involving $\vec{G}$ and $\vec{H}$ as $\vec{H}$ is composed of independently sampled elements.

Assume w.l.o.g. that the malicious Prover $\mathcal{P}^*$ knows the first m secret keys of the ring $\vec{\mathsf{pk}}$. The Prover $\mathcal{P}^*$ picks $a_1, \ldots, a_m \leftarrow\!\$\, \mathbb{Z}_q$ $b_1, \ldots, b_m \leftarrow\!\$\, \{0,1\} \subseteq \mathbb{Z}_q$ and sets $a_i = 0, b_i = 0$ for $i \in \{m+1, \ldots, n\}$. Let $\vec{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q^n$, $\vec{b} = (b_1, \ldots, b_n) \in \mathbb{Z}_2^n$.

$\mathcal{P}^*$ sets $\mathsf{cm} = e(P, L)^{\sum_{i\in[m]} a_i \mathsf{sk}_i} \cdot Q^{r_{\mathsf{cm}}}$, $\hat{A} = e(R, L)^{\sum_{i\in[m]} a_i h_i} \cdot Q^{\hat{r}_A}$, $B = e(\vec{G}, \vec{H})^{\vec{b}} \cdot Q^{r_B}$. Note that if $\mathcal{P}^*$ manages to create a verifying proof for this choice of cm, similarly to the attack of `LLRing-DL`, they can produce one unlinkable signature for every choice of $\vec{a}$.

$\mathcal{P}^*$ proceeds to run the Schnorr protocol on $\hat{A}$, and $\mathcal{P}^*$ can again create a verifying proof. Then, $\mathcal{P}^*$ sets $A = \mathsf{cm} \cdot \hat{A}$, for which holds

$$A = \mathsf{cm} \cdot \hat{A} = e(P, L)^{\sum_{i\in[m]} a_i \mathsf{sk}_i} \cdot Q^{r_{\mathsf{cm}}} \cdot e(R, L)^{\sum_{i\in[m]} a_i h_i} \cdot Q^{\hat{r}_A} = e(\vec{G}, \vec{L})^{\vec{a}} \cdot Q^{r_A}$$

and needs to find $\vec{\Omega}', \vec{\Omega}'', \vec{\Omega}''', \vec{\Theta}', \vec{\Theta}'', \vec{\Theta}'''$ such that Equation 11, Equation 12, Equation 13 hold, while setting $r'_0 = r'_1 = 0, r'_2 = r_A, r''_0 = r''_1 = r''_2 = r_B, r'''_0 = r_A, r'''_1 = r_B, r'''_2 = 0$.

Regarding Equation 11, for the equation $e(\vec{\Omega}', \vec{H}') = e(\vec{L}, \vec{H})$ to hold, $\mathcal{P}^*$ can only set $\vec{\Omega}' = \vec{L}'$ since $\vec{H}$ was selected uniformly at random from $\mathbb{G}_2^n$.

So the rest of the equations in the relation of Equation 11 become

$$e(\vec{L}', \vec{\Theta}') = e(L', L) \text{ and } e(\vec{G}, \vec{\Theta}') \cdot Q^{r_A} = A \tag{14}$$

We will find the constraints for a vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$ such that $\vec{\Theta}' = \vec{L}^{\circ\vec{x}}$ and for which Equation 14 holds.

For the equation $e(\vec{G},\vec{\Theta'})\cdot Q^{r_A}=A$, if $\mathcal{P}^*$ sets $x_i=0$ for $i\in\{m+1,\ldots,n\}$, we have $e(\vec{G},\vec{\Theta'})=\prod_{i\in[m]}e(P^{\mathsf{sk}_i}\cdot R^{h_i},L)=e(P,L)^{\sum_{i\in[m]}x_i\cdot\mathsf{sk}_i}\cdot e(R,L)^{x_i\cdot h_i}$, so given that $A=e(P,L)^{\sum_{i\in[m]}a_i\mathsf{sk}_i}\cdot e(R,L)^{\sum_{i\in[m]}a_ih_i}\cdot Q^{r_A}$, $\mathcal{P}^*$ needs to find $x_i, i\in[m]$ such that

$$e(P,L)^{\sum_{i\in[m]}x_i\cdot\mathsf{sk}_i}e(R,L)^{x_i\cdot h_i}\cdot Q^{r_A}=e(P,L)^{\sum_{i\in[m]}a_i\mathsf{sk}_i}e(R,L)^{\sum_{i\in[m]}a_ih_i}\cdot Q^{r_A}$$

which gives us the constraint

$$\begin{cases}\sum_{i\in[m]}x_i\mathsf{sk}_i=\sum_{i\in[m]}a_i\mathsf{sk}_i\\ \sum_{i\in[m]}x_ih_i=\sum_{i\in[m]}a_ih_i.\end{cases}$$

The equation $e(\vec{\Omega'},\vec{\Theta'})=e(L',L)$ is equivalent to

$$e(\vec{L'},\vec{L}^{\circ\vec{x}})=e(L',L)\Leftrightarrow e(L',L)^{\sum_{i\in[m]}x_i}=e(L',L)$$

and so we get the constraint

$$\sum_{i\in[m]}x_i=1.$$

Regarding Equation 12, $\mathcal{P}^*$ sets $\vec{\Omega''}=\vec{G}^{\circ\vec{b}}$ and $\vec{\Theta''}=\vec{H}^{\circ\vec{b}}$ (where $\vec{b}$ is the value committed in B). Notice that the equation $e(\vec{\Omega''},\vec{\Theta''})\cdot Q^{r_B}=B$ constrains $\vec{b}$ to be a binary vector, which holds by construction of $\vec{b}$.

Regarding Equation 13, for the equation $e(\vec{\Omega'''},\vec{H})\cdot Q^{r_B}=B$ to hold, $\mathcal{P}^*$ will set $\vec{\Omega'''}=\vec{G}^{\vec{b}}$, and the rest of equations of the relation of Equation 13 become

$$e(\vec{G}^{\circ\vec{b}},\vec{\Theta'''})\cdot Q^{r_A}=A \text{ and } e(\vec{G},\vec{\Theta'''})=e(\vec{G},\vec{L}). \tag{15}$$

We will again find the constraints for a vector $\vec{y}=(y_1,\ldots,y_n)\in\mathbb{Z}_q^n$ such that $\vec{\Theta'''}=\vec{L}^{\vec{y}}$ and Equation 15 holds.

$\mathcal{P}^*$ can set $y_i=1$ for $i\in\{m+1,\ldots,n\}$. So, the equation $e(\vec{G}^{\circ\vec{b}},\vec{\Theta'''})\cdot Q^{r_A}=A$ is equivalent to

$$e(\vec{G}^{\circ\vec{b}},\vec{L}^{\circ\vec{y}})\cdot Q^{r_A}=e(\vec{G}^{\circ\vec{a}},\vec{L})\cdot Q^{r_A}$$
$$\Leftrightarrow e(P,L)^{\sum_{i\in[m]}y_ib_isk_i}e(R,L)^{\sum_{i\in[m]}y_ib_ih_i}=e(P,L)^{\sum_{i\in[m]}\mathsf{sk}_ia_i}e(R,L)^{\sum_{i\in[m]}a_ih_i}$$

which gives us the constraints

$$\begin{cases}\sum_{i\in[m]}y_ib_i\mathsf{sk}_i=\sum_{i\in[m]}\mathsf{sk}_ia_i\\ \sum_{i\in[m]}y_ib_ih_i=\sum_{i\in[m]}h_ia_i\end{cases}$$

The equation $e(\vec{G},\vec{\Theta'''})=e(\vec{G},\vec{L})$ is equivalent to

$$e(\vec{G},\vec{L}^{\circ\vec{y}})=e(\vec{G},\vec{L})$$
$$\Leftrightarrow e(P,L)^{\sum_{i\in[m]y_i\mathsf{sk}_i}}e(R,L)^{\sum_{i\in[m]}y_ih_i}=e(P,L)^{\sum_{i\in[m]}\mathsf{sk}_i}e(R,L)^{\sum_{i\in[m]}h_i}$$

which gives us the constraints

$$\begin{cases} \sum_{i\in[m]} y_i \mathsf{sk}_i = \sum_{i\in[m]} sk_i \\ \sum_{i\in[m]} y_i h_i = \sum_{i\in[m]} h_i \end{cases}$$

We conclude that in order for $\mathcal{P}^*$ to create a verifying proof for $A = e(\vec{G}, \vec{L})^{\vec{a}} \cdot Q^{r_A}, B = e(\vec{G}, \vec{H})^{\vec{b}} \cdot Q^{r_B}$ with $\vec{a} \leftarrow\!\!\$\ \mathbb{Z}_q^n$ and $\vec{b} \leftarrow\!\!\$\ \{0,1\}^n$, they have to find $x_i, y_i, i \in [m]$ such that

$$\begin{cases} \sum_{i\in[m]} x_i \mathsf{sk}_i = \sum_{i\in[m]} \mathsf{sk}_i a_i \\ \sum_{i\in[m]} x_i h_i = \sum_{i\in[m]} h_i a_i \\ \sum_{i\in[m]} x_i = 1 \end{cases} \quad \begin{cases} \sum_{i\in[m]} y_i \mathsf{sk}_i = \sum_{i\in[m]} sk_i \\ \sum_{i\in[m]} y_i h_i = \sum_{i\in[m]} h_i \\ \sum_{i\in[m]} y_i b_i \mathsf{sk}_i = \sum_{i\in[m]} \mathsf{sk}_i a_i \\ \sum_{i\in[m]} y_i b_i h_i = \sum_{i\in[m]} h_i a_i \end{cases}$$

If $m > 4$, we have a system with more unknowns than equations and random coefficients, so the system has more than one solution.

4 A logarithmic verifier LRS scheme with strong linkability

In this section, we present our LRS scheme with logarithmic verification complexity.

Since our unlinkability attacks in the previous sections hinged on using $\vec{R} = (R^{h_1}, \ldots, R^{h_n})$ as a base, our first order of business is to use $\vec{R} \leftarrow\!\!\$\ \mathbb{G}_1^n$ instead. However, this change also affects the Dory proofs used by the protocol which need to be restructured. In subsection 4.2 and subsection 4.3 we present our ring membership protocol and its security analysis. Finally, in subsection 4.4 we present the full security analysis.

We note that due to space constraints, we present only the ring membership protocol of our scheme used in the Sign algorithm, since it is the protocol of `LLRing-P` we need to modify in order to avoid the unlinkability attack of section 3.3. The rest of the algorithms (the tag proof for Sign and the Link algorithm) are identical to those of `LLRing-P` as in section 3.3, and the interactive version of our scheme can be easily derived. The non-interactive version of our scheme is obtained using the Fiat-Shamir transform, and we provide the full security analysis of our non-interactive scheme in subsection 4.4.

4.1 Overview

For the ring membership proof, the prover needs to show that the commitment cm is a commitment to exactly one public key, equivalently $\mathsf{cm} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm}}} = e(\vec{\mathsf{pk}}, \vec{L}^{\circ\vec{c}}) \cdot Q^{r_{\mathsf{cm}}}$ where $\vec{c}$ is binary and sums to 1.

As in `LLRing-P`, the prover commits to $\vec{c}$ twice, in $\hat{A}$ and in B, and uses $\hat{A}$ to show that $\sum_{i\in[n]} c_i = 1$ and B to show that $\vec{c} \in \{0,1\}^n$. The differences are that

both commitments use $\vec{R} \leftarrow\!\$\, \mathbb{G}_1^n$ in place of the structured vectors of LLRing-P, and that we prove that $\vec{c}$ is binary in a different way, coordinate by coordinate. To show that $\sum_{i\in[n]} c_i = 1$, the prover commits to $\vec{c}$ via $\hat{A} = e(\vec{R}, \vec{L})^{\vec{c}} \cdot Q^{\hat{r}_A}$. The prover sets $\vec{W} = \vec{L}^{\circ\vec{c}} \in \mathbb{G}_2^n$ and proves via Dory knowledge of $\vec{W}$ committed in $\hat{A}$ such that $\prod_{i\in[n]} W_i = L$. To show that $\vec{c} \in \{0,1\}^n$, we use an additional commitment $B = e(\vec{R}, \vec{H})^{\vec{c}} \cdot Q^{r_B}$, $r_B \leftarrow\!\$\, \mathbb{Z}_q$. Setting $\vec{M} = \vec{R}^{\circ\vec{c}} \in \mathbb{G}_1^n$ and $\vec{D} = \vec{H}^{\circ\vec{c}} \in \mathbb{G}_2^n$, the commitment B can be opened in two ways,

$$B = e(\vec{M}, \vec{H}) \cdot Q^{r_B} = e(\vec{R}, \vec{D}) \cdot Q^{r_B},$$

that is, B commits to $\vec{M}$ under $\vec{H}$ and to $\vec{D}$ under $\vec{R}$. The prover proves via Dory knowledge of such $\vec{M}, \vec{D}$ for which, for every $i \in [n]$, either both $M_i = R_i$ and $D_i = H_i$ hold, or both $M_i = 1$ and $D_i = 1$ hold. Then, the prover proves again via Dory that the same vector of group elements is committed in $\hat{A}$ and cm via the fact that $\hat{A} = e(\vec{R}, \vec{W}) \cdot Q^{\hat{r}_A}$ and $\mathsf{cm} = e(\vec{\mathsf{pk}}, \vec{W}) \cdot Q^{r_{\mathsf{cm}}}$ and that $\hat{A}$ and B commit to the same $\vec{c}$ via the fact that $\hat{A} = e(\vec{M}, \vec{L}) \cdot Q^{\hat{r}_A}$ and $B = e(\vec{M}, \vec{H}) \cdot Q^{r_B}$.

To prove that for the vectors $\vec{M} \in \mathbb{G}_1^n, \vec{D} \in \mathbb{G}_2^n$ of B it holds that either $M_i = R_i$ and $D_i = H_i$ or $M_i = 1$ and $D_i = 1$, note the following. Assume that these vectors $\vec{M}, \vec{D}$ are such that for every $i \in [n]$

$$e(M_i, D_i) = e(M_i, H_i) = e(R_i, D_i). \tag{16}$$

We have that if $M_i = 1$, then $e(R_i, D_i) = e(M_i, H_i) = 1$, so $D_i = 1$. If $M_i \neq 1$, then $e(M_i, D_i H_i^{-1}) = 1$ gives $D_i = H_i$, and then $e(M_i R_i^{-1}, H_i) = 1$ gives $M_i = R_i$. So for every i, either $M_i = 1$ and $D_i = 1$, or $M_i = R_i$ and $D_i = H_i$, and the prover needs to show that Equation 16 holds for every $i \in [n]$.

To check Equation 16 for all $i \in [n]$ with only one equation, the prover sends B to the verifier, which then sends a challenge $\gamma \leftarrow\!\$\, \mathbb{Z}_q$. Let $\vec{\gamma}^n = (1, \gamma, \ldots, \gamma^{n-1})$. The prover will show that

$$e(\vec{M}, \vec{D})^{\vec{\gamma}^n} = e(\vec{M}, \vec{H})^{\vec{\gamma}^n} = e(\vec{R}, \vec{D})^{\vec{\gamma}^n} \tag{17}$$

holds.

By bilinearity, Equation 17 is equivalent to $e(\vec{M}, \vec{D}^{\circ\vec{\gamma}^n}) = e(\vec{M}, \vec{H}^{\circ\vec{\gamma}^n}) = e(\vec{R}, \vec{D}^{\circ\vec{\gamma}^n}) = e(\vec{R}^{\circ\vec{\gamma}^n}, \vec{D})$. The prover sends a hiding commitment to $\vec{D}^{\circ\vec{\gamma}^n}$,

$$B_\gamma = e(\vec{R}, \vec{D}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{B_\gamma}}.$$

If the verifier could precompute $\vec{H}^{\circ\vec{\gamma}^n}$, then the prover could provide a proof for Equation 17 with Dory that

$$e(\vec{M}, \vec{D}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{B_\gamma}} = e(\vec{M}, \vec{H}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{B_\gamma}} = e(\vec{R}, \vec{D}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{B_\gamma}} = B_\gamma,$$

using $\vec{M}, \vec{D}^{\circ\vec{\gamma}^n}$ as witnesses and $\vec{R}, \vec{H}^{\circ\vec{\gamma}^n}$ as public vectors. In a similar manner the prover can also supply a proof that B_γ is correct with respect to B, using $\vec{R}^{\circ\vec{\gamma}^n}$ as a public vector.

However, the logarithmic verifier cannot precompute $\vec{H}^{\circ\vec{\gamma}^n}, \vec{R}^{\circ\vec{\gamma}^n}$ because γ changes with every signature, and computing their commitment $e(\vec{R},\vec{H})^{\vec{\gamma}^n}$ takes n exponentiations in $\mathbb{G}_T$. So the prover additionally sends the commitment $T_\gamma = e(\vec{R},\vec{H})^{\vec{\gamma}^n}$, proves that it is correct, and will use $\vec{H}^{\circ\vec{\gamma}^n}, \vec{R}^{\circ\vec{\gamma}^n}$ as witnesses in the Dory instances.

To show that T_γ is correct, note that it commits to $\vec{H}^{\circ\vec{\gamma}^n}$ under $\vec{R}$. After sending T_γ, the prover receives a second challenge $\sigma \leftarrow\!\$\, \mathbb{Z}_q$. The prover then sends $H_{\gamma\sigma} = \prod_{i\in[n]} H_i^{(\gamma\sigma)^{i-1}}$ and shows that the vector $\vec{\Omega}$ committed in T_γ satisfies

$$\prod_{i\in[n]} \Omega_i^{\sigma^{i-1}} = H_{\gamma\sigma} = \prod_{i\in[n]} \left(H_i^{\gamma^{i-1}}\right)^{\sigma^{i-1}},$$

and that $H_{\gamma\sigma}$ is correct. Since T_γ is fixed before σ, this gives $\Omega_i = H_i^{\gamma^{i-1}}$ for every i, as for Equation 17. These two statements are instances of the relation for inner products with public vectors of scalars (see below), whose vectors of scalars are powers of σ and of $\gamma\sigma$, which have multiplicative structure, so the verifier stays logarithmic.

Dory with public vectors of scalars We use the inner-product arguments of Dory [18] for vectors of group elements committed with AFGHO commitments. Let $n = 2^m$, and let $\vec{\Gamma}_1 \leftarrow\!\$\, \mathbb{G}_1^n$ and $\vec{\Gamma}_2 \leftarrow\!\$\, \mathbb{G}_2^n$ be vectors of generators. For vectors of scalars $\vec{s}_1, \vec{s}_2 \in \mathbb{Z}_q^n$, Dory considers the relation

$$\begin{gathered}\{\vec{\Gamma}_1 \in \mathbb{G}_1^n,\ \vec{\Gamma}_2 \in \mathbb{G}_2^n,\ D_0, D_1, D_2 \in \mathbb{G}_T,\ E_1 \in \mathbb{G}_1,\ E_2 \in \mathbb{G}_2;\ \vec{V}_1 \in \mathbb{G}_1^n,\ \vec{V}_2 \in \mathbb{G}_2^n \mid \\ e(\vec{V}_1,\vec{V}_2) = D_0, \quad e(\vec{V}_1,\vec{\Gamma}_2) = D_1, \quad e(\vec{\Gamma}_1,\vec{V}_2) = D_2, \\ \prod_{i\in[n]} V_{1,i}^{s_{2,i}} = E_1, \quad \prod_{i\in[n]} V_{2,i}^{s_{1,i}} = E_2\}.\end{gathered}$$

Here D_1 and D_2 are AFGHO commitments to $\vec{V}_1$ and $\vec{V}_2$. The first three equations alone form the relation of plain Dory, which is the special case $\vec{s}_1 = \vec{s}_2 = \vec{0}$ with $E_1 = E_2 = 1$.

The verifier's precomputation depends only on the generators $\vec{\Gamma}_1$ and $\vec{\Gamma}_2$, and the vectors of scalars are folded during the protocol. If $\vec{s}_1$ and $\vec{s}_2$ have multiplicative structure, that is, $\vec{s} = \bigotimes_{i=0}^{m-1}(l_i, r_i)$ where $\otimes$ is the Kronecker product, then the folding takes $O(\log n)$ operations in $\mathbb{Z}_q$ and the verifier stays logarithmic [18, Section 4.3]. In particular, a vector of powers of a scalar $x \in \mathbb{Z}_q$ has multiplicative structure,

$$(1, x, \ldots, x^{n-1}) = (1, x^{2^{m-1}}) \otimes \cdots \otimes (1, x^2) \otimes (1, x),$$

so if $\vec{s}_1$ and $\vec{s}_2$ are vectors of powers, the verifier is logarithmic, even though x is not known in advance.

4.2 Our ring membership protocol

Setup. The setup consists of $\vec{R} \leftarrow\!\!\$\ \mathbb{G}_1^n$, $\vec{H} \leftarrow\!\!\$\ \mathbb{G}_2^n$, $(P, L') \leftarrow\!\!\$\ \mathbb{G}_1^2$, $L \leftarrow\!\!\$\ \mathbb{G}_2$, $Q \leftarrow\!\!\$\ \mathbb{G}_T$ and a hash function $\mathsf{H}_G : \{0,1\}^* \to \mathbb{G}_1$, and we set $\vec{L'} = (L', \ldots, L') \in \mathbb{G}_1^n$ and $\vec{L} = (L, \ldots, L) \in \mathbb{G}_2^n$.

Ring membership proof. The prover, with index i^* in the ring $\vec{\mathsf{pk}}$, commits to pk_{i^*} by setting $\mathsf{cm} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm}}}$ with $r_{\mathsf{cm}} \leftarrow\!\!\$\ \mathbb{Z}_q$, and proves knowledge of a witness for the ring membership relation

$$\{L \in \mathbb{G}_2,\ Q, \mathsf{cm} \in \mathbb{G}_T,\ \vec{\mathsf{pk}} \in \mathbb{G}_1^n;\ i^* \in [n],\ r_{\mathsf{cm}} \in \mathbb{Z}_q \mid \mathsf{cm} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm}}}\}.$$

The prover calculates

$$c_i = \begin{cases} 1, & \text{if } i = i^*, \\ 0, & \text{otherwise,} \end{cases}$$

samples $\hat{r}_A \leftarrow\!\!\$\ \mathbb{Z}_q$, sets

$$\hat{A} = e(\vec{R}, \vec{L})^{\vec{c}} \cdot Q^{\hat{r}_A},$$

and sends $\hat{A}$ to the verifier. In addition to cm and $\hat{A}$, the prover samples $r_B \leftarrow\!\!\$\ \mathbb{Z}_q$ and sends

$$B = e(\vec{R}, \vec{H})^{\vec{c}} \cdot Q^{r_B}.$$

The verifier sends a challenge $\gamma \leftarrow\!\!\$\ \mathbb{Z}_q$, and we write $\vec{\gamma}^n = (1, \gamma, \ldots, \gamma^{n-1})$. The prover samples $r_{B_\gamma} \leftarrow\!\!\$\ \mathbb{Z}_q$ and sends

$$B_\gamma = e(\vec{R}, \vec{H})^{(\vec{\gamma}^n \circ \vec{c})} \cdot Q^{r_{B_\gamma}}, \qquad T_\gamma = e(\vec{R}, \vec{H})^{\vec{\gamma}^n}.$$

The verifier sends a challenge $\sigma \leftarrow\!\!\$\ \mathbb{Z}_q$, and the prover sends

$$H_{\gamma\sigma} = \prod_{i \in [n]} H_i^{(\gamma\sigma)^{i-1}}.$$

Finally, the prover proves the relations $\mathcal{R}_1$–$\mathcal{R}_8$ below.

Setting $\vec{X} = \vec{\mathsf{pk}}$, $\vec{Y} = \vec{L}^{\circ\vec{c}}$, $r_{XY} = r_{\mathsf{cm}}$, $r_X = 0$ and $r_Y = \hat{r}_A$, proves

$$\begin{aligned} \mathcal{R}_1 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ Q,\ \mathsf{cm},\ e(\vec{\mathsf{pk}}, \vec{H}),\ \hat{A} \in \mathbb{G}_T; \\ &\vec{X} \in \mathbb{G}_1^n,\ \vec{Y} \in \mathbb{G}_2^n,\ r_{XY}, r_X, r_Y \in \mathbb{Z}_q \mid \\ &e(\vec{X}, \vec{Y}) \cdot Q^{r_{XY}} = \mathsf{cm}, && \text{(R1a)} \\ &e(\vec{X}, \vec{H}) \cdot Q^{r_X} = e(\vec{\mathsf{pk}}, \vec{H}), && \text{(R1b)} \\ &e(\vec{R}, \vec{Y}) \cdot Q^{r_Y} = \hat{A}\}. && \text{(R1c)} \end{aligned}$$

Setting $\vec{Z} = \vec{L'}$, $\vec{W} = \vec{L}^{\circ\vec{c}}$, $r_{ZW} = r_Z = 0$ and $r_W = \hat{r}_A$, proves

$$\begin{aligned} \mathcal{R}_2 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ Q,\ e(L', L),\ e(\vec{L'}, \vec{H}),\ \hat{A} \in \mathbb{G}_T; \\ &\vec{Z} \in \mathbb{G}_1^n,\ \vec{W} \in \mathbb{G}_2^n,\ r_{ZW}, r_Z, r_W \in \mathbb{Z}_q \mid \\ &e(\vec{Z}, \vec{W}) \cdot Q^{r_{ZW}} = e(L', L), && \text{(R2a)} \\ &e(\vec{Z}, \vec{H}) \cdot Q^{r_Z} = e(\vec{L'}, \vec{H}), && \text{(R2b)} \\ &e(\vec{R}, \vec{W}) \cdot Q^{r_W} = \hat{A}\}. && \text{(R2c)} \end{aligned}$$

Setting $\vec{\Theta} = \vec{R}$ and $\vec{\Omega} = \vec{H}^{\circ\vec{\gamma}^n}$, proves

$$\begin{aligned}\mathcal{R}_3 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ e(\vec{R},\vec{H}),\ T_\gamma \in \mathbb{G}_T,\ H_{\gamma\sigma} \in \mathbb{G}_2,\ \sigma \in \mathbb{Z}_q;\\ &\vec{\Theta} \in \mathbb{G}_1^n,\ \vec{\Omega} \in \mathbb{G}_2^n \mid \\ &e(\vec{\Theta},\vec{\Omega}) = T_\gamma, && \text{(R3a)}\\ &e(\vec{\Theta},\vec{H}) = e(\vec{R},\vec{H}), && \text{(R3b)}\\ &e(\vec{R},\vec{\Omega}) = T_\gamma, && \text{(R3c)}\\ &\prod_{i\in[n]} \Omega_i^{\sigma^{i-1}} = H_{\gamma\sigma}\}. && \text{(R3d)}\end{aligned}$$

Setting $\vec{\Lambda} = \vec{R}$ and $\vec{\Xi} = \vec{H}$, proves

$$\begin{aligned}\mathcal{R}_4 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ e(\vec{R},\vec{H}) \in \mathbb{G}_T,\ H_{\gamma\sigma} \in \mathbb{G}_2,\ \gamma,\sigma \in \mathbb{Z}_q;\\ &\vec{\Lambda} \in \mathbb{G}_1^n,\ \vec{\Xi} \in \mathbb{G}_2^n \mid \\ &e(\vec{\Lambda},\vec{\Xi}) = e(\vec{R},\vec{H}), && \text{(R4a)}\\ &e(\vec{\Lambda},\vec{H}) = e(\vec{R},\vec{H}), && \text{(R4b)}\\ &e(\vec{R},\vec{\Xi}) = e(\vec{R},\vec{H}), && \text{(R4c)}\\ &\prod_{i\in[n]} \Xi_i^{(\gamma\sigma)^{i-1}} = H_{\gamma\sigma}\}. && \text{(R4d)}\end{aligned}$$

Setting $\vec{U} = \vec{R}^{\circ\vec{c}}$, $\vec{V} = \vec{H}^{\circ(\vec{\gamma}^n \circ \vec{c})}$, $r_{UV} = r_V = r_{B_\gamma}$ and $r_U = r_B$, proves

$$\begin{aligned}\mathcal{R}_5 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ Q,\ B,\ B_\gamma \in \mathbb{G}_T;\\ &\vec{U} \in \mathbb{G}_1^n,\ \vec{V} \in \mathbb{G}_2^n,\ r_{UV}, r_U, r_V \in \mathbb{Z}_q \mid \\ &e(\vec{U},\vec{V}) \cdot Q^{r_{UV}} = B_\gamma, && \text{(R5a)}\\ &e(\vec{U},\vec{H}) \cdot Q^{r_U} = B, && \text{(R5b)}\\ &e(\vec{R},\vec{V}) \cdot Q^{r_V} = B_\gamma\}. && \text{(R5c)}\end{aligned}$$

Setting $\vec{M} = \vec{R}^{\circ\vec{c}}$, $\vec{N} = \vec{H}^{\circ\vec{\gamma}^n}$, $r_{MN} = r_{B_\gamma}$, $r_M = r_B$ and $r_N = 0$, proves

$$\begin{aligned}\mathcal{R}_6 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ Q,\ B,\ B_\gamma,\ T_\gamma \in \mathbb{G}_T;\\ &\vec{M} \in \mathbb{G}_1^n,\ \vec{N} \in \mathbb{G}_2^n,\ r_{MN}, r_M, r_N \in \mathbb{Z}_q \mid \\ &e(\vec{M},\vec{N}) \cdot Q^{r_{MN}} = B_\gamma, && \text{(R6a)}\\ &e(\vec{M},\vec{H}) \cdot Q^{r_M} = B, && \text{(R6b)}\\ &e(\vec{R},\vec{N}) \cdot Q^{r_N} = T_\gamma\}. && \text{(R6c)}\end{aligned}$$

Setting $\vec{J} = \vec{R}^{\circ\vec{\gamma}^n}$, $\vec{D} = \vec{H}^{\circ\vec{c}}$, $r_{JD} = r_{B_\gamma}$, $r_J = 0$ and $r_D = r_B$, proves

$$\begin{aligned}\mathcal{R}_7 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ Q,\ B,\ B_\gamma,\ T_\gamma \in \mathbb{G}_T;\\ &\vec{J} \in \mathbb{G}_1^n,\ \vec{D} \in \mathbb{G}_2^n,\ r_{JD}, r_J, r_D \in \mathbb{Z}_q \mid\\ &e(\vec{J}, \vec{D}) \cdot Q^{r_{JD}} = B_\gamma, &\text{(R7a)}\\ &e(\vec{J}, \vec{H}) \cdot Q^{r_J} = T_\gamma, &\text{(R7b)}\\ &e(\vec{R}, \vec{D}) \cdot Q^{r_D} = B\}. &\text{(R7c)}\end{aligned}$$

Setting $\vec{\Phi} = \vec{R}^{\circ\vec{c}}$, $\vec{\Psi} = \vec{L}$, $r_{\Phi\Psi} = \hat{r}_A$, $r_\Phi = r_B$ and $r_\Psi = 0$, proves

$$\begin{aligned}\mathcal{R}_8 = \{&\vec{R} \in \mathbb{G}_1^n,\ \vec{H} \in \mathbb{G}_2^n,\ Q,\ \hat{A},\ B,\ e(\vec{R}, \vec{L}) \in \mathbb{G}_T;\\ &\vec{\Phi} \in \mathbb{G}_1^n,\ \vec{\Psi} \in \mathbb{G}_2^n,\ r_{\Phi\Psi}, r_\Phi, r_\Psi \in \mathbb{Z}_q \mid\\ &e(\vec{\Phi}, \vec{\Psi}) \cdot Q^{r_{\Phi\Psi}} = \hat{A}, &\text{(R8a)}\\ &e(\vec{\Phi}, \vec{H}) \cdot Q^{r_\Phi} = B, &\text{(R8b)}\\ &e(\vec{R}, \vec{\Psi}) \cdot Q^{r_\Psi} = e(\vec{R}, \vec{L})\}. &\text{(R8c)}\end{aligned}$$

Relation $\mathcal{R}_1$ shows that the same vector is committed in $\hat{A}$ and cm, and $\mathcal{R}_2$ shows that the coordinates of a vector committed in $\hat{A}$ multiply to L. Relations $\mathcal{R}_3$ and $\mathcal{R}_4$ are instances of Dory with public vectors of scalars, where $\vec{\Gamma}_1 = \vec{R}$, $\vec{\Gamma}_2 = \vec{H}$ and $\vec{s}_2 = \vec{0}$. Finally, relations $\mathcal{R}_5$–$\mathcal{R}_7$ prove Equation 17, and $\mathcal{R}_8$ connects the binary vector committed in B to the vector committed in $\hat{A}$. The verifier can precompute the $\mathbb{G}_T$ elements $e(L', L)$, $e(\vec{L}', \vec{H})$, $e(\vec{R}, \vec{H})$ and $e(\vec{R}, \vec{L})$ from the setup, and $e(\vec{\mathsf{pk}}, \vec{H})$ once per ring.

4.3 Security of our ring membership protocol

We first prove some supporting lemmas that we will use in the security analysis of our ring membership protocol.

Lemma 1. *Assume DPair in $\mathbb{G}_2$. Given an argument of knowledge for $\mathcal{R}_1$, we can extract $\vec{X} \in \mathbb{G}_1^n$, $\vec{Y} \in \mathbb{G}_2^n$ and $r_{XY}, r_Y \in \mathbb{Z}_q$ such that $\vec{X} = \vec{\mathsf{pk}}$ and*

$$\mathsf{cm} = e(\vec{\mathsf{pk}}, \vec{Y}) \cdot Q^{r_{XY}}, \qquad \hat{A} = e(\vec{R}, \vec{Y}) \cdot Q^{r_Y},$$

i.e. the same vector $\vec{Y}$ is committed in both cm and $\hat{A}$.

Proof. Let $(\vec{X}, \vec{Y}, r_{XY}, r_X, r_Y)$ be the witness for $\mathcal{R}_1$ returned by the extractor. From Equation R1b we have

$$\prod_{i \in [n]} e\big(X_i \mathsf{pk}_i^{-1}, H_i\big) \cdot Q^{r_X} = 1.$$

Since $\vec{H}$ was chosen at random from $\mathbb{G}_2^n$, from DPair in $\mathbb{G}_2$ (Remark 1) we have $r_X = 0$ and $X_i \mathsf{pk}_i^{-1} = 1$, that is $X_i = \mathsf{pk}_i$, for all $i \in [n]$. Substituting $\vec{X} = \vec{\mathsf{pk}}$ into Equation R1a gives $\mathsf{cm} = e(\vec{\mathsf{pk}}, \vec{Y}) \cdot Q^{r_{XY}}$, and Equation R1c is $\hat{A} = e(\vec{R}, \vec{Y}) \cdot Q^{r_Y}$, with the same $\vec{Y}$.

Lemma 2. *Assume DPair in $\mathbb{G}_1$ and $\mathbb{G}_2$. Given an argument of knowledge for $\mathcal{R}_2$, we can extract $\vec{Z} \in \mathbb{G}_1^n$, $\vec{W} \in \mathbb{G}_2^n$ and $r_W \in \mathbb{Z}_q$ such that $\vec{Z} = \vec{L}'$ and*

$$\hat{A} = e(\vec{R}, \vec{W}) \cdot Q^{r_W}, \qquad \prod_{i \in [n]} W_i = L,$$

Proof. Let $(\vec{Z}, \vec{W}, r_{ZW}, r_Z, r_W)$ be the witness for $\mathcal{R}_2$ returned by the extractor. From Equation R2b we have

$$\prod_{i \in [n]} e\big(Z_i (L')^{-1}, H_i\big) \cdot Q^{r_Z} = 1.$$

Since $\vec{H}$ was chosen at random from $\mathbb{G}_2^n$, from DPair in $\mathbb{G}_2$ (Remark 1) we have $r_Z = 0$ and $Z_i = L'$ for all $i \in [n]$. Substituting $\vec{Z} = \vec{L}'$ into Equation R2a gives $e\big(L', \prod_{i \in [n]} W_i\big) \cdot Q^{r_{ZW}} = e(L', L)$, that is,

$$e\Big(L',\ L^{-1} \prod_{i \in [n]} W_i\Big) \cdot Q^{r_{ZW}} = 1.$$

Since L' was chosen at random from $\mathbb{G}_1$, from DPair in $\mathbb{G}_1$ (Remark 1, with $n = 1$) we have $r_{ZW} = 0$ and $\prod_{i \in [n]} W_i = L$. Finally, Equation R2c is $\hat{A} = e(\vec{R}, \vec{W}) \cdot Q^{r_W}$.

Lemma 3. *Assume DPair in $\mathbb{G}_1$. Given arguments of knowledge for $\mathcal{R}_3$ and $\mathcal{R}_4$, it holds that*

$$T_\gamma = e(\vec{R}, \vec{H}^{\circ \vec{\gamma}^n}).$$

Proof. Rewind the prover n times with n different values of σ, and assume we can extract witnesses for $\mathcal{R}_3$ and $\mathcal{R}_4$ for each such value of σ. First, notice that the witness $\vec{\Xi}$ of relation $\mathcal{R}_4$ does not depend on σ: from Equation R4c we have $e(\vec{R}, \vec{\Xi}) = e(\vec{R}, \vec{H})$ which gives $\prod_{i \in [n]} e(R_i, \Xi_i \cdot H_i^{-1}) = 1$ and from DPair in $\mathbb{G}_1$ we get that $\Xi_i = H_i$ for every $i \in [n]$. Also, notice that the witness $\vec{\Omega}$ of $\mathcal{R}_3$ does not depend on σ, since T_γ was sent before σ was given. Indeed, for two witnesses $\vec{\Omega}, \vec{\Omega}'$ for two different values of σ from Equation R3c we have $e(\vec{R}, \vec{\Omega}) = e(\vec{R}, \vec{\Omega}') \Leftrightarrow \prod_{i \in [n]} e(R_i, \Omega_i \cdot (\Omega_i')^{-1}) = 1$ and DPair in $\mathbb{G}_1$ gives $\Omega_i = \Omega_i'$ for all $i \in [n]$.

Now, substituting $\Xi_i = H_i$ into Equation R4d and combining it with Equation R3d, we get that for any such challenge σ

$$\prod_{i \in [n]} \Omega_i^{\sigma^{i-1}} = \prod_{i \in [n]} (H_i^{\gamma^{i-1}})^{\sigma^{i-1}} \tag{18}$$

Let $\sigma_1, \ldots, \sigma_n$ be the n different σ values. From Equation 18 we get

$$\begin{pmatrix} 1 & \sigma_1 & \sigma_1^2 & \cdots & \sigma_1^{n-1} \\ 1 & \sigma_2 & \sigma_2^2 & \cdots & \sigma_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_n & \sigma_n^2 & \cdots & \sigma_n^{n-1} \end{pmatrix} * \begin{pmatrix} H_1 \\ H_2^{\gamma} \\ H_3^{\gamma^2} \\ \vdots \\ H_n^{\gamma^{n-1}} \end{pmatrix} = \begin{pmatrix} 1 & \sigma_1 & \sigma_1^2 & \cdots & \sigma_1^{n-1} \\ 1 & \sigma_2 & \sigma_2^2 & \cdots & \sigma_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_n & \sigma_n^2 & \cdots & \sigma_n^{n-1} \end{pmatrix} * \begin{pmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \vdots \\ \Omega_n \end{pmatrix} \tag{19}$$

where $*$ denotes multiplication in the exponent: for a matrix $A = (a_{ki}) \in \mathbb{Z}_q^{n\times n}$ and $\vec{X} \in \mathbb{G}_2^n$, the k-th entry of $A * \vec{X}$ is $\prod_{i\in[n]} X_i^{a_{ki}}$. Since $\sigma_1, \ldots, \sigma_n$ are distinct, we can multiply both sides of Equation 19 with the inverse of the Vandermonde matrix over $\mathbb{Z}_q$, and we get that $\Omega_i = H_i^{\gamma^{i-1}}$ for all $i \in [n]$. From Equation R3c we have that $T_\gamma = e(\vec{R}, \vec{H}^{\circ\vec{\gamma}^n})$.

Lemma 4. *Assume DPair in $\mathbb{G}_1$ and $\mathbb{G}_2$, and GPI. Given arguments of knowledge for $\mathcal{R}_3$–$\mathcal{R}_7$, we can extract $\vec{M} \in \mathbb{G}_1^n$, $\vec{D} \in \mathbb{G}_2^n$ and $r_B \in \mathbb{Z}_q$ such that*

$$B = e(\vec{M}, \vec{H}) \cdot Q^{r_B} = e(\vec{R}, \vec{D}) \cdot Q^{r_B}$$

and, for every $i \in [n]$,

$$e(M_i, D_i) = e(M_i, H_i) = e(R_i, D_i),$$

i.e. the vectors $\vec{M}$ and $\vec{D}$ committed in B satisfy Equation 16.

Proof. We rewind the prover with n distinct values $\gamma_1, \ldots, \gamma_n$ of γ, keeping cm, $\hat{A}$ and B fixed. For each such value we can extract witnesses for the relations $\mathcal{R}_5$–$\mathcal{R}_7$, and Lemma 3, applied to $\mathcal{R}_3$ and $\mathcal{R}_4$, gives $T_\gamma = e(\vec{R}, \vec{H}^{\circ\vec{\gamma}^n})$. First, the witnesses $\vec{U}, \vec{M}, \vec{D}$ are the same for any value of γ, since B was sent before γ was chosen. Indeed, for two witnesses $(\vec{M}, r_M)$ and $(\vec{M}', r'_M)$ of $\mathcal{R}_6$ for two different values of γ, from Equation R6b we have $\prod_{i\in[n]} e(M_i \cdot (M'_i)^{-1}, H_i) \cdot Q^{r_M - r'_M} = 1$, and DPair in $\mathbb{G}_2$ (Remark 1) gives $\vec{M} = \vec{M}'$ and $r_M = r'_M$. In the same way, Equation R5b and DPair in $\mathbb{G}_2$ give that $\vec{U}, r_U$ do not depend on γ, and Equation R7c and DPair in $\mathbb{G}_1$ give that $\vec{D}, r_D$ do not depend on γ. Moreover, from Equations R5b and R6b we have $\prod_{i\in[n]} e(U_i \cdot M_i^{-1}, H_i) \cdot Q^{r_U - r_M} = 1$, and DPair in $\mathbb{G}_2$ gives $\vec{M} = \vec{U}$. Now, from Equation R7b we get that for every γ we have $e(\vec{J}, \vec{H}) \cdot Q^{r_J} = T_\gamma$ and from Lemma 3 we have

$$e(\vec{J}, \vec{H}) \cdot Q^{r_J} = e(\vec{R}, \vec{H}^{\circ\vec{\gamma}^n}) = e(\vec{R}^{\circ\vec{\gamma}^n}, \vec{H}), \tag{20}$$

that is, $\prod_{i\in[n]} e(J_i \cdot (R_i^{\gamma^{i-1}})^{-1}, H_i) \cdot Q^{r_J} = 1$. From DPair in $\mathbb{G}_2$ we get that $r_J = 0$ and $J_i = R_i^{\gamma^{i-1}}$ for all $i \in [n]$, meaning $\vec{J} = \vec{R}^{\circ\vec{\gamma}^n}$. Substituting into Equation R7a we get

$$e(\vec{R}^{\circ\vec{\gamma}^n}, \vec{D}) \cdot Q^{r_{JD}} = B_\gamma. \tag{21}$$

Similarly, from Equation R6c and Lemma 3 we have $\prod_{i\in[n]} e(R_i, N_i \cdot (H_i^{\gamma^{i-1}})^{-1}) \cdot Q^{r_N} = 1$, and DPair in $\mathbb{G}_1$ gives $r_N = 0$ and $\vec{N} = \vec{H}^{\circ\vec{\gamma}^n}$ for every γ. Substituting into Equation R6a gives

$$e(\vec{M}, \vec{H}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{MN}} = B_\gamma. \tag{22}$$

Now from Equations R5c and R7a with $\vec{J} = \vec{R}^{\circ\vec{\gamma}^n}$ we get that for every γ,

$$e(\vec{R}, \vec{V}) \cdot Q^{r_V} = e(\vec{R}^{\circ\vec{\gamma}^n}, \vec{D}) \cdot Q^{r_{JD}} = e(\vec{R}, \vec{D}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{JD}}. \tag{23}$$

From DPair in $\mathbb{G}_1$ we get that $r_V = r_{JD}$ and $\vec{V} = \vec{D}^{\circ\vec{\gamma}^n}$. Finally, from Equation R5a and the fact that $\vec{U} = \vec{M}$ we have

$$e(\vec{M}, \vec{D}^{\circ\vec{\gamma}^n}) \cdot Q^{r_{UV}} = B_\gamma. \tag{24}$$

That means from Equations 21, 22 and 24 we have for every $k \in [n]$,

$$e(\vec{R}^{\circ\vec{\gamma}_k^n}, \vec{D}) \cdot Q^{r_{JD,k}} = e(\vec{M}, \vec{H}^{\circ\vec{\gamma}_k^n}) \cdot Q^{r_{MN,k}} = e(\vec{M}, \vec{D}^{\circ\vec{\gamma}_k^n}) \cdot Q^{r_{UV,k}}, \tag{25}$$

where $\vec{\gamma}_k^n = (1, \gamma_k, \ldots, \gamma_k^{n-1})$ and $r_{JD,k}, r_{MN,k}, r_{UV,k}$ are the witnesses extracted for $\gamma = \gamma_k$.

We now show that the exponents of Q in Equation 25 cancel, coordinate by coordinate. Comparing the first and the third term of Equation 25, by bilinearity we have $\prod_{i\in[n]} e(R_i \cdot M_i^{-1}, D_i)^{\gamma_k^{i-1}} = Q^{r_{UV,k} - r_{JD,k}}$ for every $k \in [n]$, that is,

$$\begin{pmatrix} 1 & \gamma_1 & \cdots & \gamma_1^{n-1} \\ 1 & \gamma_2 & \cdots & \gamma_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma_n & \cdots & \gamma_n^{n-1} \end{pmatrix} * \begin{pmatrix} e(R_1 \cdot M_1^{-1}, D_1) \\ e(R_2 \cdot M_2^{-1}, D_2) \\ \vdots \\ e(R_n \cdot M_n^{-1}, D_n) \end{pmatrix} = \begin{pmatrix} Q^{r_{UV,1} - r_{JD,1}} \\ Q^{r_{UV,2} - r_{JD,2}} \\ \vdots \\ Q^{r_{UV,n} - r_{JD,n}} \end{pmatrix},$$

where $*$ denotes multiplication in the exponent: for a matrix $A = (a_{ki}) \in \mathbb{Z}_q^{n\times n}$ and $\vec{X} \in \mathbb{G}_T^n$, the k-th entry of $A * \vec{X}$ is $\prod_{i\in[n]} X_i^{a_{ki}}$. Since $\gamma_1, \ldots, \gamma_n$ are distinct, the Vandermonde matrix is invertible, and multiplying with its inverse we get $e(R_i \cdot M_i^{-1}, D_i) = Q^{\zeta_i}$ for all $i \in [n]$, where ζ_i is the i-th entry of the product of the inverse with $(r_{UV,1} - r_{JD,1}, \ldots, r_{UV,n} - r_{JD,n})^\top$. If $\zeta_i \neq 0$ for some i, then $e\big((R_i \cdot M_i^{-1})^{1/\zeta_i}, D_i\big) = Q$, which breaks GPI (4): given $z \leftarrow\!\$\ \mathbb{G}_T$, a reduction sets $Q = z$, samples the rest of the setup itself, and runs the prover and the extractor, neither of which needs the discrete logarithm of Q. So $\zeta_i = 0$ for all $i \in [n]$. Similarly, comparing the second and the third term of Equation 25, we get $e(M_i, H_i \cdot D_i^{-1}) = 1$ for all $i \in [n]$. Hence, for all $i \in [n]$,

$$e(R_i, D_i) = e(M_i, H_i) = e(M_i, D_i). \tag{26}$$

Furthermore, from Equations R6b and R7c, the vectors $\vec{M}, \vec{D}$ are committed in B: $B = e(\vec{M}, \vec{H}) \cdot Q^{r_M} = e(\vec{R}, \vec{D}) \cdot Q^{r_D}$. By Equation 26, $e(\vec{M}, \vec{H}) = \prod_{i\in[n]} e(M_i, H_i) = \prod_{i\in[n]} e(R_i, D_i) = e(\vec{R}, \vec{D})$, so $Q^{r_M} = Q^{r_D}$, that is $r_M = r_D$, and we set $r_B = r_M$.

Lemma 5. *Assume DPair in $\mathbb{G}_1$. Given an argument of knowledge for $\mathcal{R}_8$, we can extract $\vec{\Phi} \in \mathbb{G}_1^n$, $\vec{\Psi} \in \mathbb{G}_2^n$ and $r_{\Phi\Psi}, r_\Phi \in \mathbb{Z}_q$ such that $\vec{\Psi} = \vec{L}$ and*

$$\hat{A} = e(\vec{\Phi}, \vec{L}) \cdot Q^{r_{\Phi\Psi}}, \qquad B = e(\vec{\Phi}, \vec{H}) \cdot Q^{r_\Phi},$$

i.e. the same vector $\vec{\Phi}$ is committed in both $\hat{A}$ and B.

Proof. Let $(\vec{\Phi}, \vec{\Psi}, r_{\Phi\Psi}, r_\Phi, r_\Psi)$ be the witness for $\mathcal{R}_8$ returned by the extractor. From Equation R8c we have

$$\prod_{i\in[n]} e\big(R_i, \Psi_i L^{-1}\big) \cdot Q^{r_\Psi} = 1.$$

Since $\vec{R}$ was chosen at random from $\mathbb{G}_1^n$, from DPair in $\mathbb{G}_1$ (Remark 1) we have $r_\Psi = 0$ and $\Psi_i L^{-1} = 1$, that is $\Psi_i = L$, for all $i \in [n]$. Substituting $\vec{\Psi} = \vec{L}$ into Equation R8a gives $\hat{A} = e(\vec{\Phi}, \vec{L}) \cdot Q^{r_{\Phi\Psi}}$, and Equation R8b is $B = e(\vec{\Phi}, \vec{H}) \cdot Q^{r_\Phi}$, with the same $\vec{\Phi}$.

Theorem 5. *Assume DPair in $\mathbb{G}_1$ and $\mathbb{G}_2$, and GPI. If there exist arguments of knowledge for $\mathcal{R}_3$ and $\mathcal{R}_4$ and witness indistinguishable arguments of knowledge for $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_5$–$\mathcal{R}_8$, the ring membership protocol of subsection 4.2 is a witness indistinguishable argument of knowledge for the ring membership relation of Equation 7.*

Proof. **Completeness**. Completeness follows directly from construction.

CWEE. We will construct an emulator $\mathcal{E}$ that uses the emulators of the arguments of knowledge for the relations $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$–$\mathcal{R}_8$ and outputs a witness and a transcript for the ring membership relation of Equation 7.

Since there exist arguments of knowledge for $\mathcal{R}_3$–$\mathcal{R}_8$, from Lemma 4 we can extract vectors $\vec{M} \in \mathbb{G}_1^n$, $\vec{D} \in \mathbb{G}_2^n$ and $r_B \in \mathbb{Z}_q$ such that $B = e(\vec{M}, \vec{H}) \cdot Q^{r_B}$ and for all $i \in [n]$,

$$e(R_i, D_i) = e(M_i, H_i) = e(M_i, D_i).$$

Fix an $i \in [n]$. If $D_i = 1$, then we have $e(M_i, H_i) = e(R_i, 1)$ and since $H_i \neq 1$ we have $M_i = 1$. If $D_i \neq 1$, then since $e(R_i, D_i) = e(M_i, D_i)$ we have $e(R_i \cdot M_i^{-1}, D_i) = 1$ and thus $M_i = R_i$.

Let $\vec{c} = (c_1, \ldots, c_n) \in \mathbb{Z}_q^n$ be such that $c_i = 0$ if $M_i = 1$ and $c_i = 1$ if $M_i = R_i$, that is, $\vec{M} = \vec{R}^{\circ\vec{c}}$ for a binary vector $\vec{c} \in \mathbb{Z}_q^n$. We will show that $\sum_{i\in[n]} c_i = 1$.

For the witness $\vec{\Phi} \in \mathbb{G}_1^n$ and $r_\Phi \in \mathbb{Z}_q$ of $\mathcal{R}_8$, from Equation R8b and since $B = e(\vec{M}, \vec{H}) \cdot Q^{r_B}$, we get from DPair in $\mathbb{G}_2$ (Remark 1) that $\vec{\Phi} = \vec{M}$. Now, from Lemma 5 we get that $\hat{A} = e(\vec{M}, \vec{L}) \cdot Q^{\hat{r}_A}$ for $\hat{r}_A = r_{\Phi\Psi}$ and equivalently $\hat{A} = e(\vec{R}, \vec{L}^{\circ\vec{c}}) \cdot Q^{\hat{r}_A}$. Let $\vec{W} \in \mathbb{G}_2^n$, $r_W \in \mathbb{Z}_q$ be the witness for $\mathcal{R}_2$, so that $\hat{A} = e(\vec{R}, \vec{W}) \cdot Q^{r_W}$ and $\prod_{i\in[n]} W_i = L$ (Lemma 2). Comparing the two expressions for $\hat{A}$, from DPair in $\mathbb{G}_1$ we get $\vec{W} = \vec{L}^{\circ\vec{c}}$, so $\prod_{i\in[n]} L^{c_i} = L$, equivalently $\sum_{i\in[n]} c_i = 1$.

That is, $\vec{c}$ is binary and sums to 1, so it has 1 only in one position. Let that position be i^*.

Let $\vec{Y} \in \mathbb{G}_2^n$, $r_Y \in \mathbb{Z}_q$ be the witness for $\mathcal{R}_1$, so that $\hat{A} = e(\vec{R}, \vec{Y}) \cdot Q^{r_Y}$ (Lemma 1). Comparing with $\hat{A} = e(\vec{R}, \vec{L}^{\circ\vec{c}}) \cdot Q^{\hat{r}_A}$, from DPair in $\mathbb{G}_1$ we have $\vec{Y} = \vec{L}^{\circ\vec{c}}$. From Lemma 1, we have that $\mathsf{cm} = e(\vec{\mathsf{pk}}, \vec{L}^{\circ\vec{c}}) \cdot Q^{r_{XY}} = e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{XY}}$.

The emulator outputs i^* and the witness $r_{XY} \in \mathbb{Z}_q$ extracted from $\mathcal{R}_1$, together with the transcripts given by the emulators for the arguments of knowledge for $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$–$\mathcal{R}_8$ and the values $\hat{A}, B, \gamma, B_\gamma, T_\gamma, \sigma, H_{\gamma\sigma}$.

WI. Consider a statement with public value cm and witnesses $(i_1, r_{\mathsf{cm},1})$, $(i_2, r_{\mathsf{cm},2})$. For witness $(i_j, r_{\mathsf{cm},j})$, $j = 1, 2$, the prover first calculates $\vec{c}_j$ that has 1 in position i_j and 0 everywhere else, picks $\hat{r}_{A,j}, r_{B,j} \leftarrow\!\!\$\, \mathbb{Z}_q$, sets

$$\hat{A}_j = e(\vec{R}, \vec{L})^{\vec{c}_j} \cdot Q^{\hat{r}_{A,j}}, \qquad B_j = e(\vec{R}, \vec{H})^{\vec{c}_j} \cdot Q^{r_{B,j}},$$

after receiving γ picks $r_{B_\gamma,j} \leftarrow\!\!\$\ \mathbb{Z}_q$ and sets $B_{\gamma,j} = e(\vec{R},\vec{H})^{\vec{\gamma}^n \circ \vec{c}_j} \cdot Q^{r_{B_\gamma,j}}$ and $T_\gamma = e(\vec{R},\vec{H})^{\vec{\gamma}^n}$, and after receiving σ sets $H_{\gamma\sigma} = \prod_{i\in[n]} H_i^{(\gamma\sigma)^{i-1}}$. It then uses the witness indistinguishable arguments of knowledge for the relations $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_5$–$\mathcal{R}_8$ to obtain transcripts $\mathrm{tr}_{1,j}$, $\mathrm{tr}_{2,j}$ and $\mathrm{tr}_{5,j},\ldots,\mathrm{tr}_{8,j}$ respectively, and the arguments of knowledge for $\mathcal{R}_3$ and $\mathcal{R}_4$, whose witnesses $(\vec{R},\vec{H}^{\circ\vec{\gamma}^n})$ and $(\vec{R},\vec{H})$ do not depend on j, to obtain $\mathrm{tr}_{3,j}$ and $\mathrm{tr}_{4,j}$. Thus the final transcript is $(\hat{A}_j, B_j, \gamma, B_{\gamma,j}, T_\gamma, \sigma, H_{\gamma\sigma}, \mathrm{tr}_{1,j}, \mathrm{tr}_{2,j}, \mathrm{tr}_{3,j}, \ldots, \mathrm{tr}_{8,j})$.

From the hiding property of the AFGHO commitment scheme, the tuple $(\hat{A}_j, B_j, B_{\gamma,j})$ is uniform in $\mathbb{G}_T^3$ for $j = 1, 2$, and T_γ, $H_{\gamma\sigma}$, $\mathrm{tr}_{3,j}$ and $\mathrm{tr}_{4,j}$ do not depend on j. So it suffices to compare the transcripts for the same values of $\hat{A}$, B and B_γ. For these values, $\mathrm{tr}_{1,j}$, $\mathrm{tr}_{2,j}$ and $\mathrm{tr}_{5,j},\ldots,\mathrm{tr}_{8,j}$ are transcripts of the arguments for $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_5$–$\mathcal{R}_8$ on the same statements, with witnesses obtained from $(i_j, r_{\mathsf{cm},j})$. Since the arguments of knowledge satisfy witness indistinguishability, the tuples $(\mathrm{tr}_{1,1}, \mathrm{tr}_{2,1}, \mathrm{tr}_{5,1}, \ldots, \mathrm{tr}_{8,1})$ and $(\mathrm{tr}_{1,2}, \mathrm{tr}_{2,2}, \mathrm{tr}_{5,2}, \ldots, \mathrm{tr}_{8,2})$ are indistinguishable. Thus, the ring membership protocol satisfies WI.

4.4 Security analysis of our LRS scheme

Theorem 6. *The LRS scheme of section 4 satisfies correctness, anonymity, unforgeability, linkability and non-slanderability under the SXDH assumption, given that the ring membership protocol is a witness indistinguishable argument of knowledge and the tag proof protocol is a HVZK argument of knowledge.*

Before proceeding with the proof of Theorem 6, we denote two supporting lemmas that we will use in the proof.

Lemma 6 (Theorem 8 in [7]). *A linkable ring signature scheme that satisfies linkability and non-slanderability also satisfies unforgeability.*

Lemma 7 (Theorem 26 in [16]). *The tag proof protocol of Figure 2 is a perfect HVZK argument of knowledge for the tag relation of Equation 8 under the DLR assumption.*

We are now ready to prove Theorem 6. We note that since Dory is a public coin statistical HVZK argument of knowledge, by applying the Fiat-Shamir transform we obtain witness indistinguishable arguments of knowledge for $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_5$–$\mathcal{R}_8$. So by Theorem 5 the ring membership protocol is a witness indistinguishable argument of knowledge.

Proof (Proof of Theorem 6). In the proofs of anonymity and non-slanderability we will model the hash function H used to compute the linking tag as a random oracle.

Correctness Correctness follows from the completeness of the ring membership and tag proof protocols, and from the fact that for any event $f \in \mathcal{F}$ and $\mathsf{sk}_1, \mathsf{sk}_2 \in \mathbb{Z}_q$, $\mathsf{H}_G(f)^{\mathsf{sk}_1} = \mathsf{H}_G(f)^{\mathsf{sk}_2}$ if and only if $\mathsf{sk}_1 = \mathsf{sk}_2$.

Anonymity Assume an adversary $\mathcal{A}$ that manages to win the anonymity game of Definition 5 with non-negligible probability. We will construct an adversary $\mathcal{B}$ that breaks the DDH assumption in $\mathbb{G}_1$ with non-negligible probability. The adversary $\mathcal{B}$ is given as input a tuple $(X, Y, Z) \in \mathbb{G}_1$ and outputs a bit b' indicating whether it is a DDH tuple, meaning whether $X = P^a, Y = P^c, Z = P^{ac}$ for $a, c \in \mathbb{Z}_q$.

$\mathcal{B}$ plays the role of the challenger in the anonymity security game. The adversary $\mathcal{B}$ runs setup and during the key generation phase picks $j \leftarrow\$ \mathbb{Z}_q$ and sets $\mathsf{pk}_j = X$, and for $i \in [n], i \neq j$ generates the rest of the keypairs using KeyGen. Every time $\mathcal{B}$ needs to calculate $\mathsf{H}_G(f)$ for an event f, either because of a query of $\mathcal{A}$ to the random oracle or because of an $\mathcal{SO}$ query of $\mathcal{A}$, $\mathcal{B}$ acts as follows: If $\mathsf{H}_G(f)$ has not been calculated before, picks $s \leftarrow\$ \mathbb{Z}_q$, sets $\mathsf{H}_G(f) = Y^s$ and stores (f, Y^s, s). If $\mathsf{H}_G(f)$ has been calculated before, $\mathcal{B}$ returns or uses the stored value Y^s. On $\mathcal{CO}$ query of $\mathcal{A}$, if pk_j is queried, $\mathcal{B}$ halts, else $\mathcal{B}$ returns the corresponding secret key. On an $\mathcal{SO}$ query, if pk_j is queried, $\mathcal{B}$ picks $b' \leftarrow\$ \{0, 1\}$ and returns b' as the answer to the DDH game. Else $\mathcal{B}$ uses Sign to generate a signature using the corresponding secret key. Since j was chosen uniformly at random from $[n]$, the probability that pk_j is queried is $\frac{1}{n}$.

$\mathcal{A}$ will return $\mathsf{pk}_{j_0}, \mathsf{pk}_{j_1} \in [n]$ and an event f. If $j_0 \neq j$ and $j_1 \neq j$, $\mathcal{B}$ "gives up" and picks $b' \leftarrow\$ \{0, 1\}$ and returns it to the DDH game. The probability of this happening is $\frac{n-2}{n}$. Otherwise, $j = j_\tau$, for $\tau \in \{0, 1\}$. The adversary $\mathcal{B}$ picks $r_{\mathsf{cm}} \leftarrow\$ \mathbb{Z}_q$, sets $\mathsf{cm} = e(X, L) \cdot Q^{r_{\mathsf{cm}}}$ and calculates the ring membership proof using j, r_{cm} as witness. If f has been queried before to the random oracle, $\mathcal{B}$ sets $\mathsf{tag} = Z^s$, where s is the stored value corresponding to f. Else if f has not been queried before, $\mathcal{B}$ sets $\mathsf{H}_G(f) = Y$ and $\mathsf{tag} = Z$. Then, $\mathcal{B}$ uses the simulator of the tag proof to generate a proof for the statement $\mathsf{cm}, \mathsf{tag}$, and gives the final signature to $\mathcal{A}$. The adversary $\mathcal{A}$ then returns a bit $b \in \{0, 1\}$. If $b = \tau$, $\mathcal{A}$ guessed correctly indicating that the signature corresponds to public key $\mathsf{pk}_{j_\tau} = \mathsf{pk}_j = X$. Then $\mathcal{B}$ returns 1, indicating that (X, Y, Z) is a DDH tuple. Else, if $b = 1 - \tau$, $\mathcal{B}$ picks $b' \leftarrow\$ \{0, 1\}$ and returns b'.

Since j was chosen independently and in advance of the adversary's selection of $\mathsf{pk}_{j_0}, \mathsf{pk}_{j_1} \in [n]$ and because the ring membership protocol satisfies witness indistinguishability, $\mathcal{A}$ has negligible advantage in guessing b from the ring membership proof of pk_j, and we omit this advantage in the analysis. So $\mathcal{A}$ decides based on the public keys $\mathsf{pk}_{j_0}, \mathsf{pk}_{j_1} = X$ and the values $\mathsf{H}_G(f), \mathsf{tag}$, where the tuple $(\mathsf{pk}_{j_0}, \mathsf{H}_G(f), \mathsf{tag})$ is with overwhelming probability not a DDH tuple.

If (X, Y, Z) is a DDH tuple, let ϵ be the non-negligible advantage of $\mathcal{A}$ for returning $b = \tau$. Then the probability that $\mathcal{B}$ outputs 1 is

$$\begin{aligned} &1 \cdot \Pr[\mathcal{A} \text{ returns } b = \tau] + \frac{1}{2} \cdot \Pr[\mathcal{A} \text{ returns } b = 1 - \tau] \\ &= 1 \cdot \left(\frac{1}{2} + \epsilon\right) + \frac{1}{2} \cdot \left(\frac{1}{2} - \epsilon\right) = \frac{3}{4} + \frac{1}{2} \cdot \epsilon. \end{aligned}$$

If (X, Y, Z) is not a DDH tuple, then $\mathcal{A}$ has no advantage in guessing. So the probability that $\mathcal{B}$ outputs 1 is

$$1 \cdot \Pr[\mathcal{A} \text{ returns } b = \tau] + \frac{1}{2} \cdot \Pr[\mathcal{A} \text{ returns } b = 1 - \tau] = \frac{3}{4}$$

Thus, the difference between the probability that $\mathcal{B}$ returns 1 and 0 is $\frac{n-1}{n} \cdot \frac{2}{n} \cdot \frac{\epsilon}{2}$, where $\frac{n-1}{n}$ is the probability that pk_j is not queried to $\mathcal{SO}$ and $\frac{2}{n}$ is the probability that one of the indices j_0, j_1 that $\mathcal{A}$ returns is equal to j. Since ϵ is non-negligible, $\mathcal{B}$ has a non-negligible advantage in the DDH problem.

Linkability Assume an adversary $\mathcal{A}$ that manages to win the linkability game of Definition 7 with non-negligible probability. We will construct an adversary $\mathcal{B}$ that breaks the DPair assumption.

$\mathcal{B}$ is given as input $P_1, P_2 \in \mathbb{G}_1$ and plays the role of the challenger in the linkability security game. The adversary $\mathcal{B}$ runs Setup and sets $P = P_1$, $Q = e(P_2, L)$, generates the n keypairs using KeyGen, and answers the oracle queries of $\mathcal{A}$ using those keypairs. The adversary $\mathcal{A}$ returns $n+1$ unlinkable signature-message pairs $\{m_i, \sigma_i\}_{i \in [n+1]}$ and an event f. For the signatures σ_j that were an output of an $\mathcal{SO}$ query, $\mathcal{B}$ knows the corresponding witness $i_j \in [n]$, $r_{\mathsf{cm},j} \in \mathbb{Z}_q$ of the ring membership proof and the witness $\mathsf{sk}_j, \tilde{r}_{\mathsf{cm},j} \in \mathbb{Z}_q$ of the tag proof, since they provided these signatures to $\mathcal{A}$. For the rest of the signatures, $\mathcal{B}$ uses the emulators of the ring membership and tag proof protocols to extract these values.

Let $i_1, \ldots, i_{n+1}$ be the indices of the public keys for the signatures $\sigma_1, \ldots, \sigma_{n+1}$. Since we have only n public keys, at least two indices should be the same. Assume w.l.o.g. that $i_1 = i_2 = i^*$ and let $\mathsf{cm}_1, \mathsf{cm}_2$ be the commitments corresponding to σ_1, σ_2 respectively. We have

$$\begin{aligned} \mathsf{cm}_1 &= e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm},1}} = e(P, L)^{\mathsf{sk}_1} \cdot Q^{\tilde{r}_{\mathsf{cm},1}} \\ \mathsf{cm}_2 &= e(\mathsf{pk}_{i^*}, L) \cdot Q^{r_{\mathsf{cm},2}} = e(P, L)^{\mathsf{sk}_2} \cdot Q^{\tilde{r}_{\mathsf{cm},2}} \end{aligned} \tag{27}$$

Since the signatures are unlinkable, we have $\mathsf{sk}_1 \neq \mathsf{sk}_2$, since if $\mathsf{sk}_1 = \mathsf{sk}_2$ then $\mathsf{tag}_1 = \mathsf{tag}_2$ which is a contradiction. So substituting P with P_1 and Q with $e(P_2, L)$ in Equation 27 we have

$$e(P_1, L^{\mathsf{sk}_1 - \mathsf{sk}_2}) \cdot e(P_2, L^{\tilde{r}_{\mathsf{cm},1} - \tilde{r}_{\mathsf{cm},2} - r_{\mathsf{cm},1} + r_{\mathsf{cm},2}}) = 1,$$

meaning $\mathcal{B}$ is able to break the DPair assumption.

Non-slanderability Assume an adversary $\mathcal{A}$ that manages to win the non-slanderability game of Definition 8 with non-negligible probability. We construct an adversary $\mathcal{B}$ that either breaks the DLOG assumption in $\mathbb{G}_1$ or the DPair assumption.

The adversary $\mathcal{B}$ is given as input $Y, P_1, P_2 \in \mathbb{G}_1$ and will either return the DLOG of Y base P_1 or break the DPair assumption.

During setup, $\mathcal{B}$ sets $P = P_1$ and $Q = e(P_2, L)$. During the key generation phase, $\mathcal{B}$ picks $j \leftarrow\!\!\$\, [n]$ and sets $\mathsf{pk}_j = Y$, and for $i \in [n], i \neq j$ generates the rest of the public keys using KeyGen. On every hash query for an event f' of $\mathcal{A}$ to $\mathcal{B}$ that has not been asked before, $\mathcal{B}$ picks $s \leftarrow\!\!\$\, \mathbb{Z}_q$ and sets $\mathsf{H}(f') = P^s$ and stores the tuple (f', P^s, s). If f' has been asked before, $\mathcal{B}$ returns the stored value P^s.

On every query of $\mathcal{A}$ to the $\mathcal{SO}$ for a public key pk_i and event f, if $i \neq j$, $\mathcal{B}$ answers using the Sign algorithm. For pk_j, $\mathcal{B}$ first picks $r_{\mathsf{cm}} \leftarrow\!\!\$\, \mathbb{Z}_q$ and produces

a ring membership proof for $e(\mathsf{pk}_j, L) \cdot Q^{r_{\mathsf{cm}}}$ using j and r_{cm} as witness. Then, $\mathcal{B}$ calculates the ouput of $\mathsf{H}_G(f)$ and then uses the simulator of the tag proof for public input $\mathsf{cm}, \mathsf{tag} = Y^s$, where s is the value such that $\mathsf{H}_G(f) = P^s$. Notice that the simulated signature verifies. The adversary $\mathcal{A}$ returns (m, f, σ), where σ is not an output of $\mathcal{SO}$. Then, $\mathcal{B}$ uses the extractor of the ring membership and tag proofs to extract $i^*, \hat{r}_{\mathsf{cm}}$ and $\mathsf{sk}, \tilde{r}_{\mathsf{cm}}$ from the signature. If $i^* \neq j$, $\mathcal{B}$ "gives up" and returns 0. We note the probability of not giving up is non-negligible. Else, we have that

$$\mathsf{cm} = e(Y, L) \cdot Q^{\hat{r}_{\mathsf{cm}}} = e(P^{\mathsf{sk}}, L) \cdot Q^{\tilde{r}_{\mathsf{cm}}}$$

If $\hat{r}_{\mathsf{cm}} \neq \tilde{r}_{\mathsf{cm}}$, then we have $e(Y, L) \cdot e(P, L^{-\mathsf{sk}}) \cdot e(P_2, L^{\hat{r}_{\mathsf{cm}} - \tilde{r}_{\mathsf{cm}}}) = 1$, and since $P = P_1$ we have that $\mathcal{B}$ is able to break the DPair assumption. Else, if $\mathcal{B}$ did not "give up" earlier, it returns sk, the DLOG of Y base P_1.

Unforgeability From Lemma 6 we have that linkability and non-slanderability imply unforgeability.

4.5 Performance of our LRS scheme

The complexity of our logarithmic verification complexity signature scheme is dominated by the cost of running the Dory protocols. We can batch the Dory with public vectors of scalars protocols for $\mathcal{R}_3, \mathcal{R}_4$ into a single run and the zero-knowledge Dory protocols for $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_5$-$\mathcal{R}_8$ into a second single run, since they use the same generators. We thus derive a verification complexity of about $21 \log n$ and a proof size of about $15 \log n$, about 2.5 times those of `LLRing-P`.

5 Breaking the soundness of Threshold Ring Referral

The Threshold Ring Referral scheme $\Pi_{\mathsf{RR.bbs.th}}$ of Ta et al. [25] uses a similar protocol as `LLRing-P` to prove that a user possesses k BBS [6] signatures from k distinct issuer public keys, while hiding the identity of the issuer public keys in a ring. In this section we show that the attack on `LLRing-P` carries over to Threshold Ring Referral and results in a soundness attack. A malicious user, by collaborating with 3 signature issuers, can create an accepting proof for owning signatures from $k > 3$ issuers. We give an overview of the attack and we provide the full details in Appendix C.

5.1 Overview of Threshold Ring Referral

The Threshold Ring Referral scheme $\Pi_{\mathsf{RR.bbs.th}}$ consists of two parts. In the first part, the user/prover, having k signatures from k distinct issuer public keys, constructs commitments to the signatures and public keys and proves in zero knowledge the signatures verify with respect to the keys, essentially proving knowledge of k signatures from k public keys (not necessarily distinct). In the second step, the prover proves that the commitment to the keys is indeed a commitment to k distinct public keys, all belonging in a bigger ring of keys.

More precisely, the prover, having a vector of k BBS signatures $(\vec{\sigma}_1, \vec{\sigma}_2) \in \mathbb{G}_1^k \times \mathbb{Z}_q^k$ from the k public keys of the vector $\vec{\mathsf{pk}}^* = (\mathsf{pk}_{j_1}, \ldots, \mathsf{pk}_{j_k})$ with $j_i \in [n]$, computes commitments $\mathsf{cm}_{\sigma_0}, \mathsf{cm}_{\sigma_1}$ to $(\vec{\sigma}_1, \vec{\sigma}_2)$ and $\mathsf{cm}_{\mathsf{pk}}$ to $\vec{\mathsf{pk}}^*$, and proves that the committed signatures verify w.r.t. the committed keys. Then they prove that $\mathsf{cm}_{\mathsf{pk}}$ is a commitment to k distinct public keys all belonging to $\vec{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$. For this, the prover computes a vector $\vec{d} \in \mathbb{Z}_q^n$ with 1 in the positions $j \in [n]$ that the prover has a signature from public key pk_j, and 0 in all other positions. Then they commit to $\vec{d}$ by setting $\mathsf{cm}_1 = e(\vec{G}_1^{\circ \vec{d}}, \vec{K}) \cdot Q^{r_1}$, using vectors $\vec{G}_1 = (G_1, \ldots, G_1) \in \mathbb{G}_1^n$ where $G_1 \in \mathbb{G}_1$ and

$$\vec{K} = (K_1, \ldots, K_n) = (\mathsf{pk}_1 \cdot G_2^{h_1}, \ldots, \mathsf{pk}_n \cdot G_2^{h_n}) \in \mathbb{G}_2^n \tag{28}$$

with $h_i = \mathsf{H}(\mathsf{pk}_i)$ and $G_2 \in \mathbb{G}_2$. Then, the prover uses a protocol similar to that of `LLRing-P` to prove that $\vec{d}$ is a binary vector of weight k, or equivalently that $\vec{d}$ satisfies

$$\vec{d} \circ (\vec{d} - \vec{1}) = \vec{0} \text{ and } \langle \vec{d}, \vec{1} \rangle = k.$$

The prover also provides additional commitments so that the verifier can check that $\mathsf{cm}_{\mathsf{pk}}, \mathsf{cm}_1$ commit to the same set of public keys.

5.2 Attacking Threshold Ring Referral

We can show that a malicious prover, by collaborating with only 3 signature issuers, can create an accepting proof proving that they have k signatures from k distinct issuers for any $k > 3$, whilst having signatures only from those 3 issuers. The attack is based on the observation that if a prover knows some secret keys, they can compute non-trivial DLOG relations between elements of $\vec{K}$. Public keys are base G_2, so if $\mathsf{pk} = G_2^{\mathsf{sk}}$ then $\mathsf{pk} \cdot G_2^h = G_2^{\mathsf{sk}+h}$, and thus a malicious prover knowing the secret key of pk_i also knows the DLOG of K_i in base G_2.

Assume that the malicious prover $\mathcal{P}^*$ has a vector of k BBS signatures $\vec{\sigma} = (\vec{\sigma}_0, \vec{\sigma}_1) \in \mathbb{G}_1^k \times \mathbb{Z}_q^k$ and assume for simplicity that the first k_1 signatures are signed by a public key pk_1, the second k_2 from pk_2 and the last k_3 from pk_3, where $k_1 + k_2 + k_3 = k > 3$. $\mathcal{P}^*$ sets

$$\vec{\mathsf{pk}}^* = (\underbrace{\mathsf{pk}_1, \ldots, \mathsf{pk}_1}_{k_1}, \underbrace{\mathsf{pk}_2, \ldots, \mathsf{pk}_2}_{k_2}, \underbrace{\mathsf{pk}_3, \ldots, \mathsf{pk}_3}_{k_3}),$$

and computes $\mathsf{cm}_{\mathsf{pk}}$, $\mathsf{cm}_{\sigma_0}, \mathsf{cm}_{\sigma_1}$. The prover $\mathcal{P}^*$ can successfully produce a proof that the committed signatures verify w.r.t. the committed public keys, since they indeed verify.

Then, $\mathcal{P}^*$ finds $\vec{a}$ such that

$$a_i = 0 \text{ for } 3 < i < n \text{ and } \sum_{i=1}^{3} a_i(\mathsf{sk}_i + h_i) = \sum_{i=1}^{3} k_i(\mathsf{sk}_i + h_i),$$

where sk_i is the secret key corresponding to $\mathsf{pk}_i, i = 1, 2, 3$ (a simple solution is $a_i = k_i$ for $i = 1, 2, 3$), and computes $\mathsf{cm}_1 = e(\vec{G}_1^{\circ \vec{a}}, \vec{K}) \cdot Q^{r_1}$, essentially

committing to $\vec{a}$ instead of a unit vector $\vec{d}$ of weight k. Notice that the entries of $\vec{a}$ sum to k but $\vec{a}$ is not a unit vector.

The protocol to prove that $\vec{d}$ is a unit vector of weight k is similar to the protocol of `LLRing-P`, and $\mathcal{P}^*$ can apply a similar attack as the attack we described in section 3.3 to provide a verifying proof for $\vec{a}$ instead of $\vec{d}$, although $\vec{a}$ is not a unit vector. $\mathcal{P}^*$ will take advantage of the fact that they can find non-trivial DLOG relations between the 3 first elements $K_1 = G_2^{\mathsf{sk}_1+h_1}, K_2 = G_2^{\mathsf{sk}_2+h_2}, K_3 = G_2^{\mathsf{sk}_3+h_3}$ of $\vec{K}$. By requiring $\mathcal{P}^*$ to know the secret keys of more than 3 issuers, we can have more unknowns than equations in the resulting systems.

6 Discussion

In this work we identified an unlinkability vulnerability in the `LLRing` protocols and a soundness vulnerability in Threshold Ring Referral, which is caused by the use of structured group elements used in proofs. We then constructed a linkable ring signature scheme with logarithmic verification complexity and strong linkability. We note that our ring membership protocol can be applied to the scheme of Yuen et al.[29] to convert their proof to one with logarithmic verification.

The proof of `LLRing-DL` for the linkability property argues that since the range proof protocol of Bulletproofs is sound, we can use the extractor guaranteed by the soundness property to extract a witness $\vec{c}$. Thus, they do not provide a separate proof for the soundness of their modified Omniring protocol. If we could in fact guarantee extracting a vector $\vec{c}$, the linkability proof would be correct, as its argumentation is otherwise sound. As we saw however, this is not the case: successfully extracting via the soundness property is based on the hardness of finding non-trivial DLOG relations between the generators, which in their case is lacking. Similarly, the linkability proof of `LLRing-P`, claims that if $e(\vec{G}, \vec{L}^{\circ\vec{g}}) = e(\vec{G}, \vec{L})$, then $\vec{g} = \vec{1}$. As before, this is not the case, for an adversary who knows e.g. m secret keys and can find non-trivial DLOG relations for elements of $\{e(G_i, L) \mid i \in [m]\}$.

Fortunately, apart from the attack on Threshold Ring Referral, we don't identify the same attack on the (plain) Ring Referral scheme of [25]. The reason is that Ring Referral is similar to a ring signature scheme and only needs to satisfy unforgeability, in which the adversary cannot know *any* secret key of the ring. We note however that the Threshold Ring Referral scheme of [25] has no security analysis.

Finally, for our logarithmic verification complexity LRS scheme we have followed the anonymity definition of DualDory [7] and `LLRing` [15], in which the adversary sees only one signature from each of the two signers before deciding which of the two created it. Bultel and Olivier-Anclin [8] propose a stronger definition of anonymity, in which the adversary sees multiple signatures from each of the two signers before deciding, and they show that their definition is strictly stronger than the one we follow. We leave as future work proving the anonymity of our LRS scheme with respect to their stronger definition.

Disclosure Prior to submission we communicated our results to the authors of the affected papers [15,25] who have acknowledged our findings and also updated the preprint version of [25].

Errata

Previous versions of this work contained an error in the construction of Section 4 by incorrectly assuming a restricted representation of committed vectors which was not enforced by the verification equations. In this version, we repair this omission by implementing a randomization step to enforce this restriction, as well as the necessary optimizations to maintain logarithmic verification time. The fixed scheme maintains the asymptotics of the original, but in terms of constants is costlier in size and verification time by a factor of 2.5.

Acknowledgements

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30. pp. 209–236. Springer (2010)
2. Au, M.H., Chow, S.S., Susilo, W., Tsang, P.P.: Short linkable ring signatures revisited. In: European Public Key Infrastructure Workshop. pp. 101–115. Springer (2006)
3. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: logarithmic-size, no setup—from standard assumptions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 281–311. Springer (2019)
4. Balla, D., Behrouz, P., Grontas, P., Pagourtzis, A., Spyrakou, M., Vrettos, G.: Designated-verifier linkable ring signatures with unconditional anonymity. In: International Conference on Algebraic Informatics. pp. 55–68. Springer (2022)
5. Beullens, W., Katsumata, S., Pintore, F.: Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 464–492. Springer International Publishing, Cham (2020)
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Annual international cryptology conference. pp. 41–55. Springer (2004)
7. Bootle, J., Elkhiyaoui, K., Hesse, J., Manevich, Y.: DualDory: logarithmic-verifier linkable ring signatures through preprocessing. In: European Symposium on Research in Computer Security. pp. 427–446. Springer (2022)

8. Bultel, X., Olivier-Anclin, C.: On the anonymity of linkable ring signatures. In: International Conference on Cryptology and Network Security. pp. 212–235. Springer (2024)
9. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE symposium on security and privacy (SP). pp. 315–334. IEEE (2018)
10. Choi, W., Liu, X., Xia, L., Zikas, V.: K-linkable ring signatures and applications in generalized voting. Cryptology ePrint Archive (2025)
11. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the theory and application of cryptographic techniques. pp. 186–194. Springer (1986)
12. Galbraith, S.D., Hess, F., Vercauteren, F.: Aspects of pairing inversion. IEEE Transactions on Information Theory **54**(12), 5719–5728 (2008)
13. Grontas, P., Pagourtzis, A., Spyrakou, M.: Voting with coercion resistance and everlasting privacy using linkable ring signatures. Cryptology ePrint Archive (2025)
14. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 253–280. Springer (2015)
15. Hui, X., Chau, S.C.K.: LLRing: logarithmic linkable ring signatures with transparent setup. In: European Symposium on Research in Computer Security. pp. 299–319. Springer (2024)
16. Hui, X., Chau, S.C.K.: LLRing: Logarithmic linkable ring signatures with transparent setup. Cryptology ePrint Archive, Paper 2024/421 (2024). `https://doi.org/10.1007/978-3-031-70896-1_15`, `https://eprint.iacr.org/2024/421`
17. Lai, R.W., Ronge, V., Ruffing, T., Schröder, D., Thyagarajan, S.A.K., Wang, J.: Omniring: Scaling private payments without trusted setup. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 31–48 (2019)
18. Lee, J.: Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. In: Theory of cryptography conference. pp. 1–34. Springer (2021)
19. Lin, L., Qiu, T., Wang, X., Wang, H., Wei, C., Yan, Y., Wang, W., Zhao, W.: Zero-knowledge protocol for knowledge of known discrete logarithms: Applications to ring confidential transactions and anonymous zether. Cryptology ePrint Archive, Paper 2025/690 (2025), `https://eprint.iacr.org/2025/690`
20. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Linkable ring signature with unconditional anonymity. IEEE Transactions on Knowledge and Data Engineering **26**(1), 157–165 (2013)
21. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Australasian Conference on Information Security and Privacy. pp. 325–335. Springer (2004)
22. Mercer, R.: Privacy on the blockchain: Unique ring signatures. arXiv preprint arXiv:1612.01188 (2016)
23. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: International conference on the theory and application of cryptology and information security. pp. 552–565. Springer (2001)
24. Sun, S.F., Au, M.H., Liu, J.K., Yuen, T.H.: Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: European Symposium on Research in Computer Security. pp. 456–474. Springer (2017)

25. Ta, T.A., Hui, X., Chau, S.C.K.: Ring referral: Efficient publicly verifiable ad hoc credential scheme with issuer and strong user anonymity for decentralized identity and more. In: 2025 IEEE Symposium on Security and Privacy (SP). pp. 184–202. IEEE Computer Society (2025)
26. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. In: International Conference on Information Security Practice and Experience. pp. 48–60. Springer (2005)
27. Xie, M., Tu, Z., Au, M.H., Fang, J., Wang, X., Jiang, Z.L.: Efficient constant-size linkable ring signatures for ad-hoc rings via pairing-based set membership arguments. Cryptology ePrint Archive (2025)
28. Xue, Y., Lu, X., Au, M.H., Zhang, C.: Efficient linkable ring signatures: new framework and post-quantum instantiations. In: European symposium on research in computer security. pp. 435–456. Springer (2024)
29. Yuen, T.H., Chen, Y.T., Pan, S., Yu, J., Liu, J.K.: Posterior security: Anonymity and message hiding of standard signatures. Cryptology ePrint Archive (2025)
30. Yuen, T.H., Esgin, M.F., Liu, J.K., Au, M.H., Ding, Z.: Dualring: generic construction of ring signatures with efficient instantiations. In: Annual International Cryptology Conference. pp. 251–281. Springer (2021)
31. Yuen, T.H., Sun, S.f., Liu, J.K., Au, M.H., Esgin, M.F., Zhang, Q., Gu, D.: Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security. In: International Conference on Financial Cryptography and Data Security. pp. 464–483. Springer (2020)

A Additional Preliminaries

A.1 Assumptions

In this work we will be using groups in which the discrete logarithm is assumed to be hard.

Definition 10 (Group Generators). *A group generator* GGen *is a probabilistic polynomial time (p.p.t.) algorithm with input a security parameter* λ *and outputs a group description* $\mathbb{G}, g, q$ *such that* $\mathbb{G}$ *is a group of prime order* $q \approx 2^\lambda$ *with generator* g*.*

Assumption 5 (Hardness of Discrete Logarithm) *We assume that the discrete logarithm problem is hard w.r.t.* GGen*. That is, for all p.p.t* $\mathcal{A}$ *we have that*

$$\Pr[(\mathbb{G}, g, q) \leftarrow \mathsf{GGen}(1^\lambda); t \leftarrow\!\$\, \mathbb{Z}_q; h \leftarrow g^t : t = A(\mathbb{G}, g, q, h)]$$

is negligible in λ*.*

Definition 11 (Decisional Diffie-Hellman (DDH) Problem). *Let* $(\mathbb{G}, g, q) \leftarrow \mathsf{GGen}(1^\lambda)$ *be a group generated by a group generation algorithm* GGen*,*

We define the advantage $Adv^{\mathcal{A}}_{\mathsf{DDH}}$ *an algorithm* $\mathcal{A}$ *achieves advantage in the DDH problem as:*

$$Adv^{\mathcal{A}}_{\mathsf{DDH}} = \left| \Pr\left[\begin{array}{l}(\mathbb{G}, g, q) \leftarrow \mathsf{GGen}(1^\lambda);\\ a, b \leftarrow\!\$\, \mathbb{Z}_q;\\ \mathcal{A}(\mathbb{G}, g, q, g^a, g^b, g^{ab}) = 1\end{array}\right] - \Pr\left[\begin{array}{l}(\mathbb{G}, g, q) \leftarrow \mathsf{GGen}(1^\lambda);\\ a, b, c \leftarrow\!\$\, \mathbb{Z}_q;\\ \mathcal{A}(\mathbb{G}, g, q, g^a, g^b, g^c) = 1\end{array}\right] \right|$$

Assumption 6 *The DDH assumption states that for all PPT $\mathcal{A}$, $Adv^{\mathcal{A}}_{\mathsf{DDH}} = \mathsf{negl}(\lambda)$.*

A.2 Arguments of Knowledge

An *argument system* consists of three interactive algorithms $\mathsf{Setup}, \mathcal{P}, \mathcal{V}$, all running in probabilistic polynomial time. On input 1^λ, the Setup algorithm produces a common reference string pp. In the definitions below, denote by $\mathrm{tr} \leftarrow \langle \mathcal{P}(s), \mathcal{V}(t) \rangle$ the transcript produced by $\mathcal{P}$ and $\mathcal{V}$ when interacting on input s, t respectively. $\langle \mathcal{P}(s), \mathcal{V}(t) \rangle = 1$ if $\mathcal{V}$ accepts and $\langle \mathcal{P}(s), \mathcal{V}(t) \rangle = 0$ if $\mathcal{V}$ rejects.

Definition 12. *An argument system* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *is called an* argument of knowledge *for a polynomial-time decidable relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$, if it satisfies perfect completeness and computational witness extended emulation defined below.*
Prefect Completeness. *For any adversary $\mathcal{A}$,*

$$\Pr\left[\begin{matrix}(\mathsf{pp}, x, w) \notin \mathcal{R} \vee \\ \langle \mathcal{P}(\mathsf{pp}, x, w), \mathcal{V}(\mathsf{pp}, x) \rangle = 1\end{matrix} \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, w) \leftarrow \mathcal{A}(\mathsf{pp})\end{matrix}\right] = 1$$

Computational Witness Extended Emulation (CWEE). *For all deterministic polynomial-time $\mathcal{P}^*$ there exists an expected polynomial-time emulator $\mathcal{E}$ such that for all pairs of interactive adversaries $\mathcal{A}_1, \mathcal{A}_2$ we have that*

$$\left|\begin{matrix}\Pr\left[\mathcal{A}_1(tr) = 1 \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, w) \leftarrow \mathcal{A}_2(\mathsf{pp}) \\ tr \leftarrow \langle \mathcal{P}^*(\mathsf{pp}, x, w), \mathcal{V}(\mathsf{pp}, x) \rangle\end{matrix}\right] \\ - \Pr\left[\begin{matrix}\mathcal{A}_1(tr) = 1 \wedge \\ tr \text{ is accepting} \Rightarrow \\ (\mathsf{pp}, x, w) \in \mathcal{R}\end{matrix} \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, w) \leftarrow \mathcal{A}_2(\mathsf{pp}) \\ (tr, w) \leftarrow \mathcal{E}^{\mathsf{O}}(x)\end{matrix}\right]\end{matrix}\right| \leq \mathsf{negl}(\lambda),$$

where O is a rewindable oracle that can rewind the transcript $\langle \mathcal{P}^*(\mathsf{pp}, x, w), \mathcal{V}(\mathsf{pp}, x) \rangle$ and control the randomness in $\mathcal{V}$.

In the following definitions we assume pp is implicitely taken as input from both $\mathcal{P}$ and $\mathcal{V}$.

Definition 13. *An argument of knowledge* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *is called* public coin *if all messages sent from the verifier to the prover are chosen uniformly at random and independently of the prover's messages.*

Definition 14. *A public coin argument of knowledge* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *is a* statistical honest verifier zero knowledge argument of knowledge (statistical HVZK) *for $\mathcal{R}$ if there exists a probabilistic polynomial time simulator $\mathcal{S}$ such that for all pairs of interactive adversaries $\mathcal{A}_1, \mathcal{A}_2$*

$$\left| \begin{array}{l} \Pr\left[(x,w) \in \mathcal{R} \wedge \mathcal{A}_1(tr) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (x,w) \leftarrow \mathcal{A}_2(\mathsf{pp}), \\ tr \leftarrow \langle \mathcal{P}(x,w), \mathcal{V}(x) \rangle \end{array} \right] \\ - \Pr\left[(x,w) \in \mathcal{R} \wedge \mathcal{A}_1(tr) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (x,w) \leftarrow \mathcal{A}_2(\mathsf{pp}), \\ tr \leftarrow \mathsf{Sim}(x) \end{array} \right] \end{array} \right| \leq \mathsf{negl}(\lambda)$$

where r is the public coin randomness used by the verifier.

If the above difference of probabilities is equal to 0, $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *is called a* perfect HVZK *argument of knowledge for $\mathcal{R}$.*

Definition 15. *An argument of knowledge* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *for a relation $\mathcal{R}$ satisfies* witness indistinguishability *if for every probabilistic polynomial time $\mathcal{V}^*$ and every sequence (x, w, w') where $(x,w), (x,w') \in \mathcal{R}$ the following two probability ensembles are computationally indistinguishable:*

$$\{\langle \mathcal{P}(x,w), \mathcal{V}^*(x) \rangle\} \text{ and } \{\langle \mathcal{P}(x,w'), \mathcal{V}^*(x) \rangle\}$$

With the Fiat-Shamir transform, a public coin HVZK argument of knowledge also satisfies witness indistinguishability.

B Restoring the linkability of LLRing-DL

B.1 Uncompressed scheme

The goal of `LLRing-DL` is to prove that $\mathsf{cm} = \vec{\mathsf{pk}}^{\vec{c}} Q^{r_{\mathsf{cm}}}$, where $\vec{c}$ is a unit basis vector, using generators that are independent of any verifier's challenge.

On one hand, Omniring uses $\vec{G} = \vec{\mathsf{pk}}^{w} \circ \vec{R}$ where $R = (R_1, \ldots, R_n)$, and w is provided from the verifier after commiting to $\vec{c}$ using $\vec{R}$. On the other hand, as we have seen, using $\vec{R} = (R^{h_1}, \ldots, R^{h_n})$ and setting $\vec{G} = \vec{\mathsf{pk}} \circ \vec{R}$ results in an insecure protocol.

We will show that can avoid using w but also use $\vec{R} = (R_1, \ldots, R_n)$, thus also avoiding the attack.

Recall that in Omniring, the prover calculates cm and commits to $\vec{c}, \vec{c}'$ using generators $\vec{R}, \vec{H}$, i.e., sends to the Verifier the values $\mathsf{cm} = \vec{\mathsf{pk}}^{\vec{c}} Q^{r_{\mathsf{cm}}}$, $\hat{A} = \vec{R}^{\vec{c}} Q^{\hat{r}_A}$, $B = \vec{H}^{\vec{c}'} Q^{r_B}$. Then, the prover receives $w \leftarrow\!\!\$\ \mathbb{Z}_q$ from the verifier and combines cm and $\hat{A}$ to a single commitment A to $\vec{c}$ with base $\vec{G} = \vec{\mathsf{pk}}^{w} \circ \vec{R}$: $A = \mathsf{cm}^w \hat{A}$.

We can omit using A to prove that $\vec{c}$ is unit basis, and do an AND-proof to prove that $\vec{c}$ is unit basis using both $\hat{A}$ and cm. Specifically:

- We use the setup $(\vec{R}, \vec{H}, K, F, P, Q) \in \mathbb{G}^{2n+4}$ of Omniring.
- The prover calculates and sends to the verifier the values $\mathsf{cm} = \vec{\mathsf{pk}}^{\vec{c}} Q^{r_{\mathsf{cm}}}$, $\hat{A} = \vec{R}^{\vec{c}} Q^{\hat{r}_A}$, $B = \vec{H}^{\vec{c}'} Q^{r_B}$.

– The prover also picks and commits to blinding vectors $\vec{s}_1, \vec{s}_2 \leftarrow\$ \mathbb{Z}_q^n$ by sending $S_1 = \vec{\mathsf{pk}}^{\vec{s}_1} Q^{r_{S_1}}, S_2 = \vec{R}^{\vec{s}_1} Q^{r_{S_2}}, S_3 = \vec{H}^{\vec{s}_2} Q^{r_{S_3}}$.
– The prover constructs t_1, t_2, $\vec{l}(X), \vec{r}(X)$, $\hat{t}(X)$ as in Omniring/`LLRing-DL`, using verifier-provided values y, z, and evaluates $\vec{l}(X), \vec{r}(X), \hat{t}(X)$ to a random point x sent by the verifier, and sets $\vec{l} = \vec{l}(x), \vec{r} = \vec{r}(x), \hat{t} = \hat{t}(x)$.
– Finally, the prover can send $\vec{l}, \vec{r}$ to the verifier, who will calculate $\hat{t} = \langle \vec{l}, \vec{r} \rangle$ and check that

$$\begin{cases} F^{\hat{t}} Q^{\tau_x} = F^{\delta(y,z)} T_1^x T_2^{x^2} \\ \vec{\mathsf{pk}}^{\vec{l}} Q^{r_{W_1}} = \mathsf{cm} \cdot S_1^x \vec{\mathsf{pk}}^{-z \cdot \vec{1}} \\ \vec{R}^{\vec{l}} Q^{r_{W_2}} = \hat{A} S_2^x \vec{R}^{-z \cdot \vec{1}} \\ (\vec{H}^{\circ \vec{y}^{-n}})^{\vec{r}} Q^{r_{W_3}} = B S_3^x (\vec{H}^{\circ \vec{y}^{-n}})^{z \cdot \vec{y}^n + z^2 \cdot \vec{1}} \end{cases}$$

Claim. Our (linear size) LRS scheme achieves strong linkability.

Proof (sketch). The scheme contains as a sub-protocol an argument of knowledge for proving knowledge of a unit basis vector using generators $\vec{R}, \vec{H}$ chosen at random from $\mathbb{G}$. For this argument of knowledge, we can use the witness extended emulator to extract a transcript and $\vec{c}, \vec{s}_1 \in \mathbb{Z}_q^n, r_{S_1} \in \mathbb{Z}_q$ such that $\vec{l} = \vec{c} + \vec{s}_1 \cdot x - z \cdot \vec{1}$, for every evaluation point x (and a different $\vec{l}$ each time).

Take two evaluation points x_A, x_B. We have that

$$\begin{aligned} \vec{\mathsf{pk}}^{\vec{c}} \cdot \vec{\mathsf{pk}}^{x_A \cdot \vec{s}_1} \cdot Q^{r_{W_{1A}}} &= \mathsf{cm} \cdot S_2^{x_A} \\ \vec{\mathsf{pk}}^{\vec{c}} \cdot \vec{\mathsf{pk}}^{x_B \cdot \vec{s}_1} \cdot Q^{r_{W_{1B}}} &= \mathsf{cm} \cdot S_2^{x_B} \end{aligned}$$

By finding v_A, v_B such that $x_A \cdot v_A + x_B \cdot v_B = 0$ and $v_A + v_B = 1$, we have that

$$\vec{\mathsf{pk}}^{\vec{c}} Q^{v_A \cdot r_{W_{1A}} + v_B \cdot r_{W_{1B}}}$$

meaning the emulator can return the output of the sub-protocol emulator together with the value $r_{\mathsf{cm}} = v_A \cdot r_{W_{1A}} + v_B \cdot r_{W_{1B}}$. □

To compress our linear-sized signatures to logarithmic size we will use a modification of the Segregated Bulletproofs IPA (MSBP) protocol used in `LLRing-DL`, described in the next section.

This enables us to replace communicating $\vec{l}, \vec{r}$ by running the MSBP protocol for parameters

$$\{\vec{X}, \vec{Y}, \vec{H}, K, Z_1, Z_2, Z_3, \vec{a}, \vec{b}\} = \{\vec{\mathsf{pk}}, \vec{R}, \vec{H}^{\circ \vec{y}^{-n}}, K, \mathsf{cm}, \hat{A}, B, \vec{l}, \vec{r}\}$$

B.2 Modified Segregated Bulletproofs Protocol

Segregated Bulletproof IPAs, introduced in [15] use two separate commitments to carry the vectors of the inner product. We reuse their design, and modify it

Modified Segregated Bulletproofs protocol for the relation

$$\{\vec{X}, \vec{Y}, \vec{H}, K, Z_1, Z_2, Z_3; \vec{a}, \vec{b} \mid Z_1 = \vec{X}^{\vec{a}}, Z_2 = \vec{Y}^{\vec{a}}, Z_3 = \vec{H}^{\vec{b}} K^{\langle \vec{a}, \vec{b} \rangle}\}.$$

- If $n = 1$:
  $\mathcal{P} \to \mathcal{V}$: $a = \vec{a}, b = \vec{b}$
  $\mathcal{V}$: $\hat{t} = a \cdot b$, $\theta \leftarrow\$ \mathbb{Z}_q$, check that

$$X^{\theta^2 \cdot a} Y^{\theta \cdot a} H^b K^{\hat{t}} = Z_1^{\theta^2} Z_2^{\theta} Z_3$$

- Else if $n > 1$:
  $\mathcal{P}$: $\hat{t}_1 = \langle \vec{a}_R, \vec{b}_L \rangle$, $\hat{t}_2 = \langle \vec{a}_L, \vec{b}_R \rangle$
  $\mathcal{P} \to \mathcal{V}$: $L_1 = \vec{X}_L^{\vec{a}_R}, L_2 = \vec{Y}_L^{\vec{a}_R}, L_3 = \vec{H}_R^{\vec{b}_L} K^{t_1}, R_1 = \vec{X}_R^{\vec{a}_L}, R_2 = \vec{Y}_R^{\vec{a}_L}, R_3 = \vec{H}_L^{\vec{b}_R} K^{t_2}$
  $\mathcal{P} \leftarrow \mathcal{V}$: $\alpha \leftarrow\$ \mathbb{Z}_q$
  $\mathcal{P}$: $\vec{a}' = \vec{a}_L + \alpha \cdot \vec{a}_R, \vec{b}' = \alpha \cdot \vec{b}_L + \vec{b}_R$
  $\mathcal{P}\&\mathcal{V}$: $\vec{X}' = \vec{X}_L^{\alpha} \circ \vec{X}_R, \vec{Y}' = \vec{Y}_L^{\alpha} \circ \vec{Y}_R, \vec{H}' = \vec{H}_L \circ \vec{H}_R^{\alpha}$
  $\mathcal{V}$: $Z_1' = L_1^{\alpha^2} Z_1^{\alpha} R_1, Z_2' = L_2^{\alpha^2} Z_2^{\alpha} R_2, Z_3' = L_3^{\alpha^2} Z_3^{\alpha} R_3$
  $\mathcal{P}\&\mathcal{V}$: Run the protocol for input

$$(\vec{X}', \vec{Y}', \vec{H}', K, Z_1', Z_2', Z_3'; \vec{a}', \vec{b}')$$

Fig. 3. The modified Segregated Bulletproofs protocol.

slightly by also carrying over a second, redundant commitment one of the two vectors.

More concretely, we need a protocol for the relation

$$\{\vec{X}, \vec{Y}, \vec{H}, K, Z_1, Z_2, Z_3; \vec{a}, \vec{b} \mid Z_1 = \vec{X}^{\vec{a}}, Z_2 = \vec{Y}^{\vec{a}}, Z_3 = \vec{H}^{\vec{b}} K^{\langle \vec{a}, \vec{b} \rangle}\}, \quad (29)$$

in which the prover might know DLOG relations between elements of $\vec{X}$, but they don't know DLOG relations between elements of $\vec{Y}$ or $\vec{H}$.

Our modified Segregated Bulletproofs protocol appears in Figure 3. Intuitively, since we use the same $\vec{a}$ both with base $\vec{X}$ and base $\vec{Y}$, the prover is not able to cheat.

Theorem 7. *The Modified Segregated Bulletproofs protocol of Figure 3 is an argument of knowledge for the relation of Equation 29.*

Proof. Completeness follows directly.

Regarding CWEE, we will construct an emulator $\mathcal{E}$ for Modified Segregated Bulletproofs, that interacts with a prover $\mathcal{P}^*$. Notice that our Modified Segregated Bulletproofs protocol contains an instance of the Segregated Bulletproofs protocol for $\{\vec{Y}, \vec{H}, K, Z_2, Z_3; \vec{a}, \vec{b} \mid Z_2 = \vec{Y}^{\vec{a}}, Z_3 = \vec{H}^{\vec{b}} K^{\langle \vec{a}, \vec{b} \rangle}\}$ as a subprotocol. Our modification simply adds terms and checks relating to $\vec{X}$ and Z_1.

Assume that $\mathcal{P}^*$ is able to find non-trivial DLOG relations for $\vec{X}$, but they cannot find non-trivial DLOG relations for $\vec{Y}, \vec{H}$.

In the base case ($n = 1$), $\mathcal{E}$ returns the values a, b that $\mathcal{P}^*$ sends.

Assume that up to round $j-1$ we are able to extract a witness $\vec{a}', \vec{b}'$ with generators $\vec{X}', \vec{Y}', \vec{H}'$ and values Z_1', Z_2', Z_3'. For round j, we have the following.

Since Segregated Bulletproofs has CWEE, $\mathcal{E}$ can use the emulator of Segregated Bulletproofs for the statement $\{\vec{Y}, \vec{H}, K, Z_2, Z_3; \vec{a}, \vec{b} \mid Z_2 = \vec{Y}^{\vec{a}}, Z_3 = \vec{H}^{\vec{b}} K^{\langle \vec{a}, \vec{b} \rangle}\}$. The emulator of Segregated Bulletproofs returns witnesses $\vec{a}, \vec{b}$ and a transcript of interaction. We have that $Z_2 = \vec{Y}^{\vec{a}}$ and $Z_3 = \vec{H}^{\vec{b}} K^{\langle \vec{a}, \vec{b} \rangle}$, and it remains to show that $Z_1 = \vec{X}^{\vec{a}}$. The SBP emulator will rewind $\mathcal{P}^*$ with 4 different challenges $\alpha_i, i = 1, 2, 3, 4$, and for every such challenge we have that for the vector $\vec{a}'^{(i)}$ of round $j-1$ has the form $\vec{a}'^{(i)} = \vec{a}_L + \alpha_i \vec{a}_R$.

Consider the first 3 challenges. The emulator $\mathcal{E}$ can find $v_i, i = 1, 2, 3$ such that

$$\sum_{i=1,2,3} v_i \alpha_i = 1, \qquad \sum_{i=1,2,3} v_i \alpha_i^2 = 0, \qquad \sum_{i=1,2,3} v_i = 0.$$

Then, since for $j-1$ the transcript verifies, we have that for $i = 1, 2, 3$,

$$\begin{aligned} Z_1'^{(i)} = \vec{X}^{\vec{a}'^{(i)}} = L_1^{\alpha_i^2} Z_1^{\alpha_i} R_1 \Rightarrow (\vec{X}_L^{\alpha_i} \circ \vec{X}_R)^{\vec{a}_L + \alpha_i \cdot a_R} = L_1^{\alpha_i^2} Z_1^{\alpha_i} R_1 \\ \Rightarrow \vec{X}_L^{\alpha_i \cdot \vec{a}_L} \vec{X}_L^{\alpha_i^2 \cdot \vec{a}_R} \vec{X}_R^{\alpha_i \cdot \vec{a}_R} \vec{X}_R^{\vec{a}_L} = L_1^{\alpha_i^2} Z_1^{\alpha_i} R_1. \end{aligned}$$

Combining these equations with v_1, v_2, v_3 we get that $\vec{X}^{\vec{a}} = Z_1$. So $\mathcal{E}$ returns $\vec{a}, \vec{b}$ and the transcript given by the SBP emulator.

B.3 Efficiency

Restoring the linkability of `LLRing-DL` is dominated by the cost of running modified Segregated Bulletproofs. The proof size is $6 \log n$, and the prover and the verifier compute $3n$ multi-exponentiations. By setting $\vec{H} = \vec{R}$ we can reduce the number of multi-exponentiations to $2n$. The restoration results in a worse performing protocol than Omniring, which also requires $2n$ multi-exponentiations but has a proof size of $2 \log n$, however, it might be of independent interest.

C Full attack on Threshold Ring Referral

In this section we show the full attack to the protocol $\Pi_{\mathsf{RR.bbs.th}}$ of [25]. The attack uses a similar reasoning as the attack of `LLRing`. A malicious prover, by collaborating with only 3 signature issuers, can create an accepting proof for any $k > 3$, while only having signatures from these 3 issuers. We need 3 issuers so that we have more unknowns than equations in the resulting systems.

In the $\Pi_{\mathsf{RR.bbs.th}}$ protocol, we have the following vectors

- $\vec{K} = (K_1, \ldots, K_n) = (\mathsf{pk}_1 \cdot G_2^{h_1}, \ldots, \mathsf{pk}_n \cdot G_2^{h_n}) \in \mathbb{G}_2^n$ with $h_i = \mathsf{H}(\mathsf{pk}_i)$ and $G_2 \in \mathbb{G}_2$, and public keys are base G_2, so if $\mathsf{pk} = G_2^{\mathsf{sk}}$ then $\mathsf{pk} \cdot G_2^h = G_2^{\mathsf{sk}+h}$.

– $\vec{\Gamma} \in \mathbb{G}_1^n, \vec{\Gamma}' = (\vec{\Gamma}, 1_{\mathbb{G}_1}) \in \mathbb{G}_1^{n+1}, \vec{\Omega} \in \mathbb{G}_1^n, \vec{\Lambda} \in \mathbb{G}_2^n, \vec{\Lambda}' = (\vec{\Lambda}, 1_{\mathbb{G}_2}) \in \mathbb{G}_2^{n+1}$ are all vectors of random generators.
– $\vec{G}_1 = (G_1, \ldots, G_1) \in \mathbb{G}_1^k$ and $\vec{G}_2 = (G_2, \ldots, G_2) \in \mathbb{G}_2^k$.

Assume that the malicious prover $\mathcal{P}^*$ has k_1, k_2, k_3 signatures from issuers with public keys $\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{pk}_3$, respectively such that $k_1 + k_2 + k_3 = k > 3$. If the vector of signatures is $\vec{\sigma} = (\vec{\sigma_0}, \vec{\sigma_1})$, assume for simplicity that the first k_1 signatures are signed by pk_1, the second k_2 from pk_2 and the last k_3 from pk_3.

$\mathcal{P}^*$ sets

$$\vec{\mathsf{pk}}^* = (\underbrace{\mathsf{pk}_1, \ldots, \mathsf{pk}_1}_{k_1}, \underbrace{\mathsf{pk}_2, \ldots, \mathsf{pk}_2}_{k_2}, \underbrace{\mathsf{pk}_3, \ldots, \mathsf{pk}_3}_{k_3}) \tag{30}$$

$$\vec{h}^* = (\underbrace{h_1, \ldots, h_1}_{k_1}, \underbrace{h_2, \ldots, h_2}_{k_2}, \underbrace{h_3, \ldots, h_3}_{k_3}) \tag{31}$$

$$\vec{K}^* = (\underbrace{K_1, \ldots, K_1}_{k_1}, \underbrace{K_2, \ldots, K_2}_{k_2}, \underbrace{K_3, \ldots, K_3}_{k_3}) \tag{32}$$

$\mathcal{P}^*$ creates $\mathsf{cm}_{\sigma_0}, \mathsf{cm}_{\sigma_1}, \mathsf{cm}_{\mathsf{pk}}, \mathsf{cm}_{\mathsf{m}}$ honestly (in $\mathsf{cm}_{\mathsf{pk}}$ he uses $\vec{\mathsf{pk}}^*$ of Equation 30), and runs $\Pi_{\mathsf{bbs.pms.th}}$ to produce a proof that those k signatures verify. The prover $\mathcal{P}^*$ is able to create a verifying proof, since the signatures indeed verify.

Then, $\mathcal{P}^*$ finds $\vec{a}$ such that

$$a_i = 0 \text{ for } 3 < i < n \text{ and } \sum_{i=1}^{3} a_i(\mathsf{sk}_i + h_i) = \sum_{i=1}^{3} k_i(\mathsf{sk}_i + h_i),$$

where n is the ring size and sk_i is the secret key corresponding to $\mathsf{pk}_i, i = 1, 2, 3$.

$\mathcal{P}^*$ sets $\mathsf{cm}_1 = e(\vec{G}_1^{\vec{a}}, \vec{K}) \cdot Q^{r_1}$ and creates $\mathsf{cm}_2, \mathsf{cm}_h, R$ using $\vec{K}^*, \vec{h}^*$ of Equation 30. This means that $\mathsf{cm}_2 = e(\vec{\Gamma}, \vec{K}^*) \cdot Q^{r_2}$ and $\mathsf{cm}_h = \vec{\Gamma}^{\vec{h}^*} \cdot \Gamma^{r_D}$.

Notice that in the final verification step, the verifier will check an equation corresponding to the check $e(\vec{\Gamma}, \vec{K}^*) = e(\vec{\Gamma}, \vec{\mathsf{pk}}^*) \cdot e(\vec{\Gamma}, \vec{G}_2^{\circ \vec{h}^*})$. Given the construction of $\mathsf{cm}_2, \mathsf{cm}_{\mathsf{pk}}$, the equation will be satisfied. So it remains to show that $\mathcal{P}^*$ can create a verifying proof for $\Pi^{\mathsf{th}}_{\mathsf{chkuv}}, \Pi_{\mathsf{do.ip}}$.

In $\Pi^{\mathsf{th}}_{\mathsf{chkuv}}$, $\mathcal{P}^*$ sets $\mathsf{cm}'' = e(\vec{\Omega}^{\vec{b}}, \vec{K}) \cdot Q^{r''}$ with[7] $b = (1, 1, 0, \ldots 0) \in \mathbb{Z}_q^n$. Then, $\mathcal{P}^*$ has to find witnesses $\vec{X}_1, \vec{Y}_2, \vec{X}_2, \vec{Y}_2, \vec{X}_3, \vec{Y}_3$ for the following relations.

$$\begin{gathered}\{\vec{\Omega} \in \mathbb{G}_1^n, \vec{K} \in \mathbb{G}_2^n, Q, e(G_1, G_2)^k, \mathsf{cm}_1, e(\vec{\Omega}, \vec{G}_2) \in \mathbb{G}_T; \vec{X}_1 \in \mathbb{G}_1^n, \vec{Y}_1 \in \mathbb{G}_2^n, r_1 \in \mathbb{Z}_q \mid \\ e(\vec{X}_1, \vec{Y}_1) = e(G_1, G_2)^k, \\ e(\vec{X}_1, \vec{K}) \cdot Q^{r_1} = \mathsf{cm}_1, \\ e(\vec{\Omega}, \vec{Y}_1) = e(\vec{\Omega}, \vec{G}_2)\}\end{gathered} \tag{33}$$

[7] Note that $b_3 = 0$.

$$\{\vec{\Omega} \in \mathbb{G}_1^n, \vec{K} \in \mathbb{G}_2^n, Q, \mathsf{cm}'' \in \mathbb{G}_T; \vec{X}_2 \in \mathbb{G}_1^n, \vec{Y}_2 \in \mathbb{G}_2^n, r'' \in \mathbb{Z}_q \mid \begin{array}{c} e(\vec{X}_2, \vec{Y}_2) \cdot Q^{r''} = \mathsf{cm}'', \\ e(\vec{X}_2, \vec{K}) \cdot Q^{r''} = \mathsf{cm}'', \\ e(\vec{\Omega}, \vec{Y}_2) \cdot Q^{r''} = \mathsf{cm}''\} \end{array} \tag{34}$$

$$\{\vec{\Omega} \in \mathbb{G}_1^n, \vec{K} \in \mathbb{G}_2^n, Q, \mathsf{cm}_1, e(\vec{G}_1, \vec{K}), \mathsf{cm}'' \in \mathbb{G}_T; \vec{X}_3 \in \mathbb{G}_1^n, \vec{Y}_3 \in \mathbb{G}_2^n, r_1, r'' \in \mathbb{Z}_q \mid \begin{array}{c} e(\vec{X}_3, \vec{Y}_3) \cdot Q^{r_1} = \mathsf{cm}_1, \\ e(\vec{X}_3, \vec{K}) = e(\vec{G}_1, \vec{K}), \\ e(\vec{\Omega}, \vec{Y}_3) \cdot Q^{r''} = \mathsf{cm}''\} \end{array} \tag{35}$$

Consider the relation of Equation 33. From the constraint $e(\vec{\Omega}, \vec{Y}_1) = e(\vec{\Omega}, \vec{G}_2)$, since $\vec{\Omega}$ is a vector of random generators, we have that $\vec{Y}_1 = \vec{G}_2$.

Let $\vec{X}_1 = \vec{G}_1^{\circ \vec{x}_1}$. The prover $\mathcal{P}^*$ can set $x_{3i} = 0$ for $3 < i < n$.

Now from the constraint $e(\vec{X}_1, \vec{Y}_1) = e(G_1, G_2)^k$, we have

$$e(\vec{G}_1^{\circ \vec{x}_1}, \vec{G}_2) = e(G_1, G_2)^k \Leftrightarrow \prod_{i=1,2,3} e(G_1^{x_{1i}}, G_2) = e(G_1, G_2)^k$$
$$\Leftrightarrow e(G_1, G_2)^{\sum_{i=1,2,3} x_{1i}} = e(G_1, G_2)^k \Leftrightarrow \sum_{i=1,2,3} x_{1i} = k.$$

From the constraint $e(\vec{X}_1, \vec{K}) \cdot Q^{r_1} = \mathsf{cm}_1$, we have

$$e(\vec{G}_1^{\circ \vec{x}_1}, \vec{K}) = e(\vec{G}_1^{\circ \vec{a}}, \vec{K}) \Leftrightarrow \prod_{i=1,2,3} e(G_1^{x_{i1}}, G_2^{\mathsf{sk}_i + h_i}) = \prod_{i=1,2,3} e(G_1^{a_i}, G_2^{\mathsf{sk}_i + h_i})$$
$$\Leftrightarrow e(G_1, G_2)^{\sum_{i=1,2,3} x_{i1}(\mathsf{sk}_i + h_i)} = e(G_1, G_2)^{\sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i)}$$
$$\Leftrightarrow \sum_{i=1,2,3} x_{i1}(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i).$$

Consider the relation of Equation 34. If $\vec{X}_2 = \vec{\Omega}^{\circ \vec{x}_2}$ and $\vec{Y}_2 = \vec{K}^{\circ \vec{y}_2}$, we have that $\vec{x}_2 = \vec{y}_2 = \vec{b}$ and $b_i \in \{0, 1\}$, which holds by construction of $\vec{b}$.

Regarding the relation of Equation 35, from the constraint $e(\vec{\Omega}, \vec{Y}_3) \cdot Q^{r''} = \mathsf{cm}''$ we have $\vec{Y}_3 = \vec{K}^{\circ \vec{b}}$. If $\vec{X}_3 = \vec{G}_1^{\circ \vec{x}_3}$, the prover $\mathcal{P}^*$ can set $x_{3i} = 0$ for $3 < i < n$. From the constraint $e(\vec{X}_3, \vec{K}) = e(\vec{G}_1, \vec{K})$, we have

$$e(\vec{G}_1^{\circ \vec{x}_3}, \vec{K}) = e(\vec{G}_1, \vec{K}) \Leftrightarrow \prod_{i=1,2,3} e(G_1^{x_{3i}}, G_2^{\mathsf{sk}_i + h_i}) = \prod_{i=1,2,3} e(G_1, G_2^{\mathsf{sk}_i + h_i})$$
$$\Leftrightarrow e(G_1, G_2)^{\sum_{i=1,2,3} x_{3i}(\mathsf{sk}_i + h_i)} = e(G_1, G_2)^{\sum_{i=1,2,3} \mathsf{sk}_i + h_i}$$
$$\Leftrightarrow \sum_{i=1,2,3} x_{3i}(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} \mathsf{sk}_i + h_i$$

From the constraint $e(\vec{X}_3, \vec{Y}_3) \cdot Q^{r_1} = \mathsf{cm}_1$, we have

$$e(\vec{G}_1^{\circ\vec{x}_3}, \vec{K}^{\circ\vec{b}}) = e(\vec{G}_1^{\circ\vec{a}}, \vec{K}) \Leftrightarrow \prod_{i=1,2,3} e(G_1^{x_{3i}}, G_2^{b_i(\mathsf{sk}_i+h_i)}) = \prod_{i=1,2,3} e(G_1^{a_i}, G_2^{\mathsf{sk}_i+h_i})$$
$$\Leftrightarrow e(G_1, G_2)^{\sum_{i=1,2,3} x_{3i}b_i(\mathsf{sk}_i+h_i)} = e(G_1, G_2)^{\sum_{i=1,2,3} a_i(\mathsf{sk}_i+h_i)}$$
$$\Leftrightarrow \sum_{i=1,2,3} x_{3i}b_i(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i)$$
$$\Leftrightarrow \sum_{i=1,2} x_{3i}(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i)$$

So for $\mathcal{P}^*$ to create a verifying proof for $\Pi^{\mathsf{th}}_{\mathsf{chkuv}}$, they have to find $x_{1i}, x_{3i}, i = 1, 2, 3$ such that

$$\begin{cases} \sum_{i=1,2,3} x_{1i} = k \\ \sum_{i=1,2,3} x_{1i}(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i) \end{cases}$$
$$\begin{cases} \sum_{i=1,2,3} x_{3i}(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} \mathsf{sk}_i + h_i \\ \sum_{i=1,2} x_{3i}(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i) \end{cases}$$

This is easily achievable as the left hand side system has 2 equations for 3 unknowns x_{11}, x_{12}, x_{13}, and, independently the right side system again has 2 equations for 3 unknowns x_{31}, x_{32}, x_{33}. We also note that the latter of the RHS equations omits x_{33} as b_3 was set to 0.

Then, $\mathcal{P}^*$ will have to create a verifying proof for $\Pi_{\mathsf{do.ip}}$, meaning they have to find a witness $\vec{X}, \vec{Y}$ for the following relation.

$$\{\vec{\Gamma}' \in \mathbb{G}_1^{n+1}, \vec{\Lambda}' \in \mathbb{G}_2^{n_1}, Q, \mathsf{cm}_1, e(\vec{G}_1, \vec{\Lambda}), \mathsf{cm}_2 \in \mathbb{G}_T; \vec{X} \in \mathbb{G}_1^{n+1}, \vec{Y} \in \mathbb{G}_2^{n+1}, r_1, r_2 \in \mathbb{Z}_1 \mid$$
$$e(\vec{X}, \vec{Y}) \cdot Q^{r_1} = \mathsf{cm}_1,$$
$$e(\vec{X}, \vec{\Lambda}') = e(\vec{G}_1, \vec{\Lambda}),$$
$$e(\vec{\Gamma}', \vec{Y}) \cdot Q^{r_2} = \mathsf{cm}_2\}$$

That is, they equivalently have to find $\vec{X}, \vec{Y}$ that satisfy the following equations.

$$e(\vec{X}, \vec{Y}) = e(\vec{G}_1^{\circ\vec{a}}, \vec{K}) \qquad e(\vec{X}, \vec{\Lambda}') = e(\vec{G}_1, \vec{\Lambda}) \qquad e(\vec{\Gamma}', \vec{Y}) = e(\vec{\Gamma}, \vec{K}^*).$$

From the second and third equation, $\mathcal{P}^*$ should set $\vec{Y} = (\vec{K}^*, 1_{\mathbb{G}_2}) = \vec{K}^{*\prime}, \vec{X} = (\vec{G}_1, 1_{\mathbb{G}_1}) = \vec{G}_1'$. So, the first equation becomes

$$e(\vec{G}_1', \vec{K}^{*\prime}) = e(\vec{G}_1^{\circ\vec{a}}, \vec{K}) \Leftrightarrow \prod_{i=1,2,3} e(G_1, K_i^{k_i}) = \prod_{i=1,2,3} e(G_1^{a_i}, K_i)$$
$$\Leftrightarrow e(G_1, G_2)^{\sum_{i=1,2,3} k_i(\mathsf{sk}_i+h_i)} = e(G_1, G_2)^{\sum_{i=1,2,3} a_i(\mathsf{sk}_i+h_i)}$$
$$\Leftrightarrow \sum_{i=1,2,3} k_i(\mathsf{sk}_i + h_i) = \sum_{i=1,2,3} a_i(\mathsf{sk}_i + h_i),$$

which holds by the construction of $\vec{a}$.