A note on the Unsuitability of LIGA for Linkable Ring Signatures

The perils of non-commutativity

Iñigo Diaz Iribarnegaray
University of Bordeaux
inigo.diaz-iribarnegaray@etu.u-bordeaux.fr

Václav Gregor
University of Bordeaux
vaclav.gregor@aquilenet.fr

Florian L'écu Leal
University of Bordeaux
florianleculeal@gmail.com

Abstract. In this work, we study the proposal for a linkable ring signature (LRS) in [KTS+24]. It is instantiated from the group action based framework described in [BKP20], using the Lattice Isomorphism Group Action (LIGA), meaning that the security of the signature rests on the famous Lattice Isomorphism Problem (LIP).

We will show that this signature does *not* in fact fit the requirements to be a linkable ring signature, despite the guarantees of the [BKP20] framework, due to it straying from that framework by using a non-commutative group for the group actions. More specifically, we will show that the signature from [KTS+24] satisfies neither the property of *correctness* nor *linkability*, which are required of a LRS.

This further damages the signature, as it was already shown in [BCF25] that the *linkable anonymity* property of [KTS+24] isn't satisfied.

The group used in LIGA is the group of invertible integer matrices: $\mathrm{GL}_n(\mathbb{Z})$. As the main obstacle in successfully applying the framework mentioned above to construct a LRS based on LIP is the fact that this group is non-commutative, we try fixing the signature by restricting the secret key space to a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$. However, we will see that finding a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$ that maintains the hardness of the underlying LIP, and that at the same time is realistic to use is not as easy as it may seem.

April 6, 2026

Contents

1 Introduction **3**

2 Preliminaries **4**
- 2.1 Notations . . . 4
- 2.2 Lattice Isomorphism Problem . . . 4
  - 2.2.1 Basis Lattice Isomorphism . . . 4
  - 2.2.2 Quadratic Form Lattice Isomorphism . . . 5
- 2.3 Group Actions and LIGA . . . 6
  - 2.3.1 Group Actions . . . 6
  - 2.3.2 Cryptographic Group Actions . . . 6
  - 2.3.3 Lattice Isomorphism Group Action . . . 7
- 2.4 (Linkable) Ring Signatures . . . 7
  - 2.4.1 Ring Signatures . . . 7
  - 2.4.2 Linkable Ring Signatures . . . 9

3 Group Action Based Linkable Ring Signature **10**
- 3.1 General construction . . . 10
- 3.2 LIGA instantiation . . . 12

4 LIGA-LRS is not a Linkable Ring Signature **12**
- 4.1 LIGA-LRS is not Correct . . . 12
- 4.2 LIGA-LRS is not Linkable . . . 13

5 Attempting to correct the Signature **14**
- 5.1 Actions of the form $\mathbf{U} \bullet \mathbf{Q} := f(\mathbf{U}) \star \mathbf{Q}$ . . . 14
  - 5.1.1 Necessary condition for Linkable Anonymity . . . 14
  - 5.1.2 Sufficient condition for Linkability . . . 14
- 5.2 Commutative subgroups of $\mathrm{GL}_n(\mathbb{Z})$ . . . 15
  - 5.2.1 Groups generated by a single matrix . . . 15
  - 5.2.2 Shear matrices . . . 16
  - 5.2.3 Polynomials of a nilpotent matrix . . . 16

References **19**

We would like to thank our tutor, *Alice Pellet-Mary*, researcher at the University of Bordeaux, for her guidance and support during this project.

1 Introduction

With the advent of quantum computing on the way, migration to post-quantum cryptographic algorithms becomes ever more important. This is not only true of basic primitives such as KEMs and DSAs, but also of more *fancy* cryptography, such as *linkable ring signatures* (LRS).

Ring signatures (predecessor of *linkable* ring signatures) were first introduced by Rivest, Shamir, and Tauman in 2001 [RST01]. Given a set of people called a *ring*, a ring signature allows a member of the ring to sign a message, in a way where anyone can verify that the signature comes from within the ring, but no one is able to determine which member of the ring signed the message. A linkable ring signature is simply an extension of a ring signature, where we can also verify that two signatures come from the same person, without finding out the identity of the signer, thus the word *linkable*. This is achieved by including an object called the *tag* along with each signature. If two signatures share the same tag (or have tags that are similar enough), we conclude that they come from the same person.

In this work, we study the proposal for a linkable ring signature from [KTS+24], which we will refer to as *LIGA-LRS* throughout our paper. It rests on the hardness of the Lattice Isomorphism Problem (LIP), which is used to instantiate the group action based framework described in [BKP20] using the Lattice Isomorphism Group Action (LIGA).

In [BCF25], Budroni, Chi-Domínguez, and Franch have already shown that this signature doesn't satisfy the property of *linkable anonymity*, which is one of the properties required of LRS. Linkable anonymity roughly translates by the infeasibility of finding out the identity of the signer, given a signature and its tag. In their case, the attack has nothing to do with the general framework from [BKP20], but rather comes from LIGA directly, as the authors show that LIGA isn't *2-weakly pseudorandom*, which results in an attack against linkable anonymity of the signature. Intuitively, it means that given a public-key+tag pair, one can distinguish this pair from a random one, which gives an attacker the ability to deduce which member of the ring was at the origin of the signature, thus breaking linkable anonymity.

Our contribution is more focused on the general framework, and on the way the LRS from [KTS+24] was instantiated using this framework. We show that LIGA-LRS does not satisfy *correctness* nor *linkability*. Coupled with the attack from [BCF25], this means that 3 of the properties required of linkable ring signatures are not satisfied by LIGA-LRS, rendering it unusable.

The correctness and linkability aren't satisfied by LIGA-LRS, despite the guarantees of the [BKP20] framework, due to it straying from that framework by using a non-commutative group for the group actions: $\mathrm{GL}_n(\mathbb{Z})$, the group of invertible integer matrices.

Our attempts at correcting the signature thus consist of swapping out $\mathrm{GL}_n(\mathbb{Z})$ for a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$ in the definition of LIGA, which would make LIGA-LRS align with the general framework. Based on this, the security properties of the signature should be guaranteed by the framework itself, *as long as the underlying problem (LIP) stays hard*. As we will see at the end of this paper, finding a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$ that is realistic to use and that keeps the hardness of LIP, let alone the 2-weakly pseudorandomness of LIGA, is not evident. We mention 3 candidates for such subgroup, going from a naive construction to a more sophisticated one. For each one, we explain why the construction fails, hoping that this preliminary analysis might help future research.

We must however highlight that none of these issues are present in the non-linkable version of the signature, as it only needs one group action, and LIP seems perfectly suited for this task.

Our work is organized as follows: Section 2 will provide the necessary notions for working with the signature. Section 3 will present the signature as proposed in [KTS+24], within the framework of [BKP20]. Section 4 will show that the signature lacks *corectness* and *linkability*. Of these proofs, two of them are of our own, while one is directly from [BCF25]. Section 5 will present our attempts to remedy the issues with the signature, to no avail.

2 Preliminaries

We start by setting the notations we will use all along the document. Then, we introduce preliminary mathematical and cryptographic notions. More precisely, we focus on the Lattice Isomorphism Problem (LIP), which is the algorithmic problem the construction we will study is based on. We also introduce group actions, their applications in cryptography and their link with LIP. All this leads us to the definition of the Lattice Isomorphism Group Action (LIGA). We also define ring signatures (the predecessor of LRS) and linkable ring signatures in this section. These notions will be helpful for the study of the specific linkable ring signature from [KTS+24], the main object of this document.

2.1 Notations

- Matrices are denoted in bold uppercase letters, for example $\mathbf{M}$. In particular, the identity matrix is denoted $\mathbf{I}_n$, and the zero matrix is denoted $\mathbf{0}_n$.
- Vectors are denoted in bold lowercase letters, for example $\mathbf{v}$.
- Lattices are denoted with calligraphic $\mathcal{L}$'s. We abuse this notation and use the letter $\mathcal{L}$ to denote the lattice generated by some basis as well. For example, in $\mathcal{L} = \mathcal{L}(\mathbf{B})$, $\mathcal{L}$ is the name of the lattice generated by the basis $\mathbf{B}$.
- $\mathbb{Z}^{n\times n}$: the ring of $n \times n$ matrices over $\mathbb{Z}$.
- $\mathrm{GL}_n(\mathbb{Z})$: the group of $n \times n$ unimodular matrices over $\mathbb{Z}$, that is invertible integer matrices with determinant ± 1.
- $\mathcal{O}_n(\mathbb{R})$: orthonormal $n \times n$ matrices over $\mathbb{R}$.
- $\mathcal{S}_n^{>0}(\mathbb{R})$: symmetric and positive-defnite $n \times n$ matrices over $\mathbb{R}$.
- In general, anything starting with $\mathcal{D}$ is a distribution.

2.2 Lattice Isomorphism Problem

Two lattices $\mathcal{L}$ and $\mathcal{L}'$ given by their respective bases $\mathbf{B}$ and $\mathbf{B}'$ are identical if and only if there exists a unimodular matrix $\mathbf{U}$ such that $\mathbf{B} = \mathbf{B}' \cdot \mathbf{U}$. Finding such $\mathbf{U}$ (if it exists) is algorithmically easy, and therefore not suited for cryptographic use.

Fortunately, when we zoom out a little and consider the isomorphism of lattices instead, we get a much harder problem. We formulate it in terms of full-rank lattices, as those are the lattices that we will be using. The definitions of this section are the standard lattice isomorphism definitions, as found for example in [BCF25].

2.2.1 Basis Lattice Isomorphism

Intuitively, two lattices are isomorphic if one can be obtained from the other by rotation and/or reflection, without stretching or skewing. More precisely, the isomorphism has to be a linear transformation that preserves the inner product, thus corresponding to some orthonormal matrix. The following definition captures this idea, describing the isomorphism as first changing the basis by multiplying it by some unimodular matrix $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$, then applying the orthonormal matrix $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ to obtain a basis of an isomorphic lattice.

Definition 1 (Lattice Isomorphism)**.** *Two n-dimensional full-rank lattices $\mathcal{L} = \mathcal{L}(\mathbf{B})$ and $\mathcal{L}' = \mathcal{L}(\mathbf{B}') \subset \mathbb{R}^n$ are said to be* isomorphic*, when there exists a unimodular matrix $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ and an orthonormal matrix $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\mathbf{B} = \mathbf{O} \cdot \mathbf{B}' \cdot \mathbf{U}$.*

We can now formulate a corresponding algorithmic problem.

Definition 2 (Lattice Isomorphism Problem (LIP) – basis version)**.** *Given the bases* $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{n\times n}$ *of two isomorphic lattices, find* $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ *and* $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ *such that* $\mathbf{B} = \mathbf{O} \cdot \mathbf{B}' \cdot \mathbf{U}$*.*

However, because of the real-valued $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, this formulation of LIP is hard to implement in practice [DW22, Section 2.2]. We can get around this problem by reformulating LIP in terms of *quadratic forms*.

2.2.2 Quadratic Form Lattice Isomorphism

Definition 3 (Quadratic Form)**.** *We define a* quadratic form *as a symmetric and positive-definite* $n \times n$ *matrix over* $\mathbb{R}$*. We denote the set of quadratic forms of size* $n \times n$ *as* $\mathcal{S}_n^{>0}(\mathbb{R})$*.*

For $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{n\times n}$ bases of two isomorphic lattices, consider the quadratic forms $\mathbf{Q}$ and $\mathbf{Q}'$ defined as $\mathbf{Q} := \mathbf{B}^\intercal\mathbf{B}$ and $\mathbf{Q}' := \mathbf{B}'^\intercal\mathbf{B}'$. One can easily verify that $\mathbf{Q}$ and $\mathbf{Q}'$ are actually symmetric and positive-definite, which comes from the fact that $\mathbf{B}$ and $\mathbf{B}'$ are full-rank.

We can now reformulate LIP in terms of these matrices.

$$\begin{aligned} \text{Since } \mathbf{B} &= \mathbf{O} \cdot \mathbf{B}' \cdot \mathbf{U}, \\ \text{then, } \mathbf{Q} = \mathbf{B}^\intercal\mathbf{B} &= (\mathbf{O}\mathbf{B}'\mathbf{U})^\intercal \mathbf{O}\mathbf{B}'\mathbf{U} \\ &= \mathbf{U}^\intercal\mathbf{B}'^\intercal\mathbf{O}^\intercal\mathbf{O}\mathbf{B}'\mathbf{U} \\ &= \mathbf{U}^\intercal\mathbf{B}'^\intercal\mathbf{B}'\mathbf{U} \\ &= \mathbf{U}^\intercal\mathbf{Q}'\mathbf{U}, \end{aligned}$$

since $\mathbf{O}^\intercal\mathbf{O} = \mathbf{I}_n$ because $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$. This leads us to the following definition.

Definition 4 (Equivalence of Quadratic Forms)**.** *We say that two quadratic forms* $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R})$ *are* equivalent*, when there exists* $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ *such that*

$$\mathbf{Q}' = \mathbf{U}^\intercal\mathbf{Q}\mathbf{U}.$$

Note that this is a symmetric relation, as we can also write $\mathbf{Q} = \mathbf{U}^{-\intercal}\mathbf{Q}'\mathbf{U}^{-1}$ *with* $\mathbf{U}^{-1} \in \mathrm{GL}_n(\mathbb{Z})$*. We denote* $[\mathbf{Q}]$ *the equivalence class of* $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ *under this relation, that is*

$$[\mathbf{Q}] := \left\{\mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R}) \mid \exists \mathbf{U} \in \mathrm{GL}_n(\mathbb{Z}), \mathbf{Q}' = \mathbf{U}^\intercal\mathbf{Q}\mathbf{U}\right\}$$

All this allows us to reformulate LIP in terms of quadratic forms.

Definition 5 (Lattice Isomorphism Problem (LIP) – quadratic form version)**.** *Given two equivalent quadratic forms* $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ *and* $\mathbf{Q}' \in [\mathbf{Q}]$*, find* $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ *such that* $\mathbf{Q}' = \mathbf{U}^\intercal\mathbf{Q}\mathbf{U}$*. We then call* $(\mathbf{Q}, \mathbf{Q}')$ *a* LIP instance*.*

We mentioned before that the main motivation for reformulating LIP in terms of quadratic forms was to get rid of the real-valued orthonormal $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, yet we defined our quadratic forms as real-valued matrices. To clarify the difference between the two, suppose we have a basis $\mathbf{B} \in \mathbb{R}^{n\times n}$ of some full-rank lattice we want to work with. In the context of the basis formulation of LIP, if we want to talk about lattices isomorphic to $\mathcal{L}(\mathbf{B})$, we are forced to consider real-valued $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ at some point. On the other hand, in the quadratic form formulation of LIP, we can describe all isomorphisms of $\mathcal{L}(\mathbf{B})$ simply with integer matrices $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$. Furthermore, since in practical applications $\mathbf{B}$ is integer-valued, $\mathbf{Q} := \mathbf{B}^\intercal\mathbf{B}$ is as well, which makes the computations and representations easier. Overall, we see that the quadratic forms formulation offers more flexibility and an easier implementation. However, since one might want to work with non-integer lattices anyway, we keep things as general as possible and define quadratic forms as real matrices, only restricting them to be integer valued when actually necessary.

LIP is widely believed to be a difficult problem. For example, the signature HAWK [BBD+25], a candidate in the ongoing NIST post-quantum signature competition, is based on an algebraic variant of LIP.

2.3 Group Actions and LIGA

In 1990, Brassard and Yung took notice of the prevalence of group actions and group-action-like constructions with cryptographic applications, unifying them under the notion of *one-way* group actions [BY90]. Since then, cryptographic group actions have been used to construct many advanced schemes, in particular the ones discussed in this paper.

2.3.1 Group Actions

We start this section by recalling the definition of a group action.

Definitions 1 (Left/Right Group Action)**.** *Given a group $(G, \cdot, e)$ and a set X, a function $\star : G \times X \to X$ is a* left group action *if it satisfies the following properties.*

- Identity*:* $\forall x \in X : e \star x = x$.
- Compatibility*:* $\forall g, h \in G, \forall x \in X : g \star (h \star x) = (g \cdot h) \star x$.

Given a group $(G, \cdot, e)$ and a set X, a function $\star : G \times X \to X$ is a right group action *if it satisfies the following properties.*

- Identity*:* $\forall x \in X : e \star x = x$.
- Compatibility*:* $\forall g, h \in G, \forall x \in X : g \star (h \star x) = (h \cdot g) \star x$.

Left group actions are often simply called group actions, which is fine for commutative groups, since in that case there is no distinction between left and right group actions. However, in this paper it is crucial that we distinguish between them, given that we will be discussing actions of non-commutative groups.

When talking about group actions without specifying left or right, we are referring to both left and right group actions. We may sometimes shorten group action to action, and left/right group action to left/right action.

2.3.2 Cryptographic Group Actions

In order to do cryptography with group actions, we need them to hold some additional properties. On one hand, we need to be able to handle the group action efficiently, on the other, we need some specific operations related to our group action to be hard. First of all, we want the ability to manipulate our group action easily. More formally, as first presented in [Cou06] and further formalized in [ADM+20], we require that group actions be *effective*:

Definition 6 (Effective Group Action)**.** *A group action $\star : G \times X \to X$ is* effective *if there exists a fixed origin element $x_0 \in X$ of known bit representation and PPT algorithms to*

- *test whether a given bit string is an element of G or X,*
- *sample $g \in G$ from some distribution $\mathcal{D}_G$,*
- *for $g, h \in G, x \in X$, compute gh, g^{-1}, and $g \star x$,*
- *for $g_1, g_2 \in G$ test whether $g_1 = g_2$,*
- *for $x_1, x_2 \in X$ test whether $x_1 = x_2$.*

In what follows, we will require all of our group actions to be effective. As for the cryptographic properties, we will be discussing the following ones, which were first defined in [BY90; Cou06] and further formalized in [ADM+20; BBC+24]:

Definitions 2 (Cryptographic Properties of Group Actions)**.** *A group action $\star : G \times X \to X$ is said to be:*

- One-way, *if there is no PPT algorithm that given $x, x' \in X$ finds $g \in G$ such that $g \star x = x'$ with non-negligible probability.*
- t-weakly unpredictable, *if there is no PPT algorithm that given t pairs of the form $(x_i, x_i' := g \star x_i) \in X \times X$ for some $g \in G$, and an additional $y \in X$ different from all x_i's, outputs $y' := g \star y$ with non-negligible probability.*
- t-weakly pseudorandom *for a given distribution $\mathcal{D}_X$, if there is no PPT algorithm that can distinguish between t pairs of the form $(x_i, x_i' := g \star x_i) \in X \times X$ for some $g \in G$ and $x_i \leftarrow \mathcal{D}_X$, and t random pairs $(x_i, x_i') \leftarrow \mathcal{D}_X \times \mathcal{D}_X$, with non-negligible advantage.*

2.3.3 Lattice Isomorphism Group Action

Definition 7 (LIGA)**.** *Let $\mathbf{Q}_0 \in \mathcal{S}_n^{>0}(\mathbb{R})$ be a quadratic form and $[\mathbf{Q}_0]$ its equivalence class. The* Lattice Isomorphism Group Action (LIGA) *$\star$ is defined as:*

$$\begin{aligned} \star : \mathrm{GL}_n(\mathbb{Z}) \times [\mathbf{Q}_0] &\to [\mathbf{Q}_0] \\ (\mathbf{U}, \mathbf{Q}) &\mapsto \mathbf{U} \star \mathbf{Q} := \mathbf{U}^\intercal \mathbf{Q} \mathbf{U} \end{aligned}$$

We denote the set of automorphisms of a quadratic form $\mathbf{Q}$ under LIGA as $\mathrm{Aut}(\mathbf{Q}) := \{\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z}) \mid \mathbf{U} \star \mathbf{Q} = \mathbf{Q}\}$.

In practical applications (such as signatures), $\mathbf{Q}_0$ is taken integer-valued from $\mathcal{S}_n^{>0}(\mathbb{Z})$. In that case, in what follows we will often be using a Gaussian distribution over the equivalence class $[\mathbf{Q}_0]$ of parameter s, denoted $\mathcal{D}_s([\mathbf{Q}_0])$, and defined in [DW22, Definition 3.3]. As long as

$$s \geqslant \max\left\{\lambda_n(\mathbf{Q}_0), \|\mathbf{B}^*_{\mathbf{Q}_0}\|\sqrt{\ln(2n+4)/\pi}\right\},$$

in [DW22, Lemma 3.4] the authors prove that there exists a polynomial-on-average time algorithm that samples $\mathbf{Q}$ from $\mathcal{D}_s([\mathbf{Q}_0])$ and also produces a $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ uniform among those satisfying $\mathbf{Q} = \mathbf{U} \star \mathbf{Q}_0$. We will denote by $\mathcal{D}_{\mathrm{GL}_n(\mathbb{Z})}$ the resulting distribution on $\mathrm{GL}_n(\mathbb{Z})$ from this algorithm when s is taken equal to the lower bound of [DW22, Lemma 3.4].

2.4 (Linkable) Ring Signatures

This subsection is dedicated to ring signatures and linkable ring signatures, which are extensions of the usual digital signatures.

2.4.1 Ring Signatures

Ring signatures (RS) were first introduced by Rivest, Shamir, and Tauman in 2001 [RST01]. The proposed motivation behind them is to allow whistle-blowers to leak sensitive information from within an organization, in a way where anyone can verify the authenticity of the information, but no one can deduce which member of the organization was at the origin of the leak.

To achieve this, the only thing that the whistle-blower needs is the public signature keys of other members of the organization. The whistle-blower can then use these public keys along with their own secret key to construct a ring signature used to sign the leak. Anyone can verify that the signer was in possession of one of the secret keys of the organization, but no one, not even other members of the organization, can trace the signature back to the whistle-blower. This idea is illustrated in figure 1.

The set of public keys of the members of the organization is called a *ring*. The member producing the signature (the whistle-blower) is called the *signer*. All the other members are called *non-signers*.

In the original whistle-blowing context, it is crucial that the ring signature take as input actual public signature keys of the ring members, as otherwise no one would believe the signer that the leak is legitimate. As an example, this means that if the management of the company

in which our whistle-blower operates uses ECDSA to sign their messages, the ring signature has to work with ECDSA public keys.

In our work however, we concentrate on a more modern vision of ring signatures, which have found applications other than those of leaking secrets, for instance in electronic vote protocols. As such, we often simplify the scenario by assuming that the public keys of the ring members can be used directly in the ring signature, without modifying them in some special way. In other words, we willingly forget the part where the ring signature's construction would be conditioned by the actual public keys used by the organization.

The following definition is a standard modern definition of a ring signature, as stated in a paper on constructing (linkable) ring signatures from isogenies and lattices by Beullens, Katsumata, and Pintore [BKP20].

Definition 8 (Ring Signature)**.** *Given a message space $\mathcal{M}$, a signature space Σ and a security parameter λ, a* ring signature *is defined by the following PPT algorithms.*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$. *Given the security parameter λ, produces public parameters* pp.
- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{pk})$. *Given the public parameters* pp*, produces a secret/public key pair* $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{Sign}(\mathcal{R} = (\mathsf{pk}_1, \mathsf{pk}_2, \ldots, \mathsf{pk}_N), \mathsf{sk}, m \in \mathcal{M}) \to \sigma$. *Given a ring of public keys $\mathcal{R}$, the signer's secret key* sk*, and a message m, produces the ring signature $\sigma \in \Sigma$.*
- $\mathsf{Verify}(\mathcal{R} = (\mathsf{pk}_1, \mathsf{pk}_2, \ldots, \mathsf{pk}_N), \sigma \in \Sigma, m \in \mathcal{M}) \to \{0, 1\}$. *Given the ring of public keys $\mathcal{R}$, a signature σ, and a message m, outputs 1 if σ is a valid ring signature of m for this ring and 0 otherwise.*

We also require that the ring signature satisfy the following properties. For formal definitions of these properties, we refer the reader to [BKP20, Definition 2.2].

- Correctness. *Verifying a correctly generated signature always succeeds.*
- Full Anonymity. *It is infeasible to find out which secret key was used to produce the signature, even if the adversary knows the secret and public keys of all ring members.*
- Unforgeability. *It is infeasible to produce a valid signature without knowing a secret key corresponding to one of the public keys of the ring.*

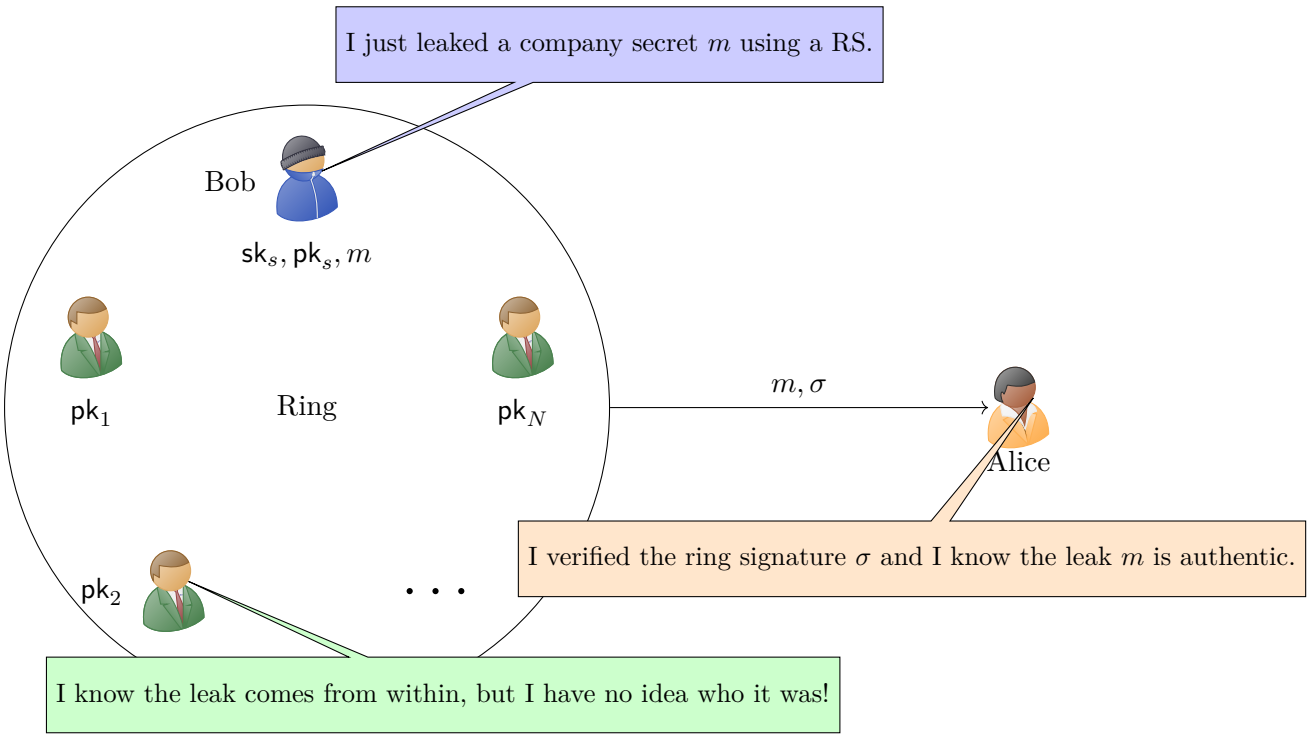

Figure 1: Use case of a ring signature.

2.4.2 Linkable Ring Signatures

A *linkable* ring signature is simply an extension of a ring signature, where we can also verify whether two signatures were produced by the same signer, without finding out the identity of the signer. This is done through the means of an object called the *tag*. Each member of the ring has their own tag which they send along with their signatures. If the tags of two signatures coincide, we know they come from the same signer. We once again use the definition from [BKP20, Definition 2.3], leaving out the formal parts for reasons of brevity.

Definition 9 (Linkable Ring Signature)**.** *A* linkable ring signature *is a ring signature with the following additional PPT algorithm.*
$\mathsf{Link}(\sigma_1 \in \Sigma, \sigma_2 \in \Sigma) \to \{0,1\}$. *Given two signatures σ_1 and σ_2, outputs 1 if these signatures were produced with the same secret key and 0 otherwise.*

In addition to this algorithm, we require the following properties for linkable ring signatures, in lieu of those of non-linkable ring signatures.

- Correctness. *Verifying a correctly generated signature always succeeds and $\mathsf{Link}(\sigma, \sigma) = 1$ for all signature σ.*
- Linkability. *It is infeasible to produce two valid signatures σ_1, σ_2 corresponding to the same public key, such that $\mathsf{Link}(\sigma_1, \sigma_2) = 0$.*
- Linkable Anonymity. *It is infeasible for an adversary knowing the public keys of the ring members to find out which public key corresponds to a given signature. In contrast to* anonymity *of ring signatures, here the adversary is not given the secret keys, as they could use the* linkability *property to deduce who produced the signature.*
- Non-frameability. *Given a valid signature σ, it is infeasible for an adversary not in possession of the secret key used to produce σ to forge a signature $\sigma' \neq \sigma$, such that $\mathsf{Link}(\sigma, \sigma') = 1$.*

- Unforgeability. *Same as for ring signatures. It can be shown that unforgeability is already implied by linkability and non-frameability [BKP20, Remark 2.4], but we include it anyway for completeness.*

These definitions given, we can now study how to use group actions to build a specific kind of linkable ring signature.

3 Group Action Based Linkable Ring Signature

As explained before, we now focus on a particular type of linkable ring signatures, namely *Group Action Based Linkable Ring Signatures*. We first study their construction presented in [BKP20] through a simplified version that presents none of the optimizations, looking only at the underlying sigma protocol. Then we look at how to use LIGA in this context, instantiating the construction as done in [KTS+24].

3.1 General construction

In [BKP20], Beullens, Katsumata, and Pintore introduce a general method for defining linkable ring signatures based on group actions. Given a commutative[1] group G acting on two (potentially equal) sets $\mathcal{X}, \mathcal{T}$ by two effective actions $\star : G \times \mathcal{X} \to \mathcal{X}$ and $\bullet : G \times \mathcal{T} \to \mathcal{T}$, and an efficiently computable function $\mathsf{Link}_{\mathsf{GA}} : \mathcal{T} \to \{0, 1\}$, the authors provide a general method for constructing a linkable ring signature. The idea behind the pair of actions is that from fixed public parameters $X_0 \in \mathcal{X}, T_0 \in \mathcal{T}$ and a secret key $s \in G$, a member of the ring may construct their public key as $X := s \star X_0$. The linkable aspect of the signature is then achieved with the second action, as the member of the ring may construct the tag $T := s \bullet T_0$. The tag can then be used when signing, using $\mathsf{Link}_{\mathsf{GA}}$ to check whether two signatures were produced by the same member of the ring, without revealing the associated public key, and thus maintaining anonymity. According to [BKP20], the reason $\mathsf{Link}_{\mathsf{GA}}$ is used instead of just checking whether the tags from two signatures are equal is to allow for the use of computational problems such as LWE, where the tags may not be strictly equal, but just close according to some metric.

If $(\star, \bullet, \mathsf{Link}_{\mathsf{GA}})$ forms an *admissible pair of group actions* as defined in [BKP20], then the construction satisfies all properties required to be a linkable ring signature. Here we give a simplified formulation of this admissibility taken from [BCF25]. The reader will notice that the names of the conditions are identical to those of the definition of linkable ring signatures. This is intentional, as each condition on $(\star, \bullet, \mathsf{Link}_{\mathsf{GA}})$ is necessary for the respective condition on the constructed linkable ring signature.

Definition 10 (Admissible pair of group actions)**.** *Given* $(\star, \bullet, \mathsf{Link}_{\mathsf{GA}})$ *as described above, we say it is an* admissible pair of group actions *with respect to* $X_0 \in \mathcal{X}$, $T_0 \in \mathcal{T}$, *and a distribution* $\mathcal{D}_G$ *on the group* G, *if the following conditions hold:*

- Correctness. $\forall T \in \mathcal{T}, \mathsf{Link}_{\mathsf{GA}}(T, T) = 1$.
- Linkability. *It is infeasible to find* $s, s' \in G$ *such that* $s \star X_0 = s' \star X_0$ *but* $\mathsf{Link}_{\mathsf{GA}}(s \bullet T_0, s' \bullet T_0) = 0$.
- Linkable Anonymity. *The distributions*

$$\{(s \star X_0, s \bullet T_0 \mid s \leftarrow \mathcal{D}_G\}$$

and

$$\{(s \star X_0, s' \bullet T_0) \mid s \leftarrow \mathcal{D}_G, s' \leftarrow \mathcal{D}_G\}$$

are indistinguishable.

[1] The commutativity requirement of G is only mentioned once in [BKP20], and its necessity is never justified, making it very easy to disregard. However, as we will later see, it is actually quite important for the construction to work as presented.

- Non-frameability. *Given* $X = s \star X_0$ *and* $T = s \bullet T_0$ *it is infeasible to find* s' *such that* $\mathsf{Link}_{\mathsf{GA}}(T, s' \bullet T_0) = 1$.

The signature scheme centers around a group action sigma protocol/proof of knowledge, based on the sigma protocol for graph isomorphisms [GMW86]. The sigma protocol is adapted into an OR sigma protocol for the anonymity of the ring signature, and then it is adapted into an OR sigma protocol proving the knowledge of a secret s such that $s \star X_0$ is a public key of the ring, and $s \bullet T_0$ is the signature's tag, thus allowing linkability. As is customary, this binary-challenge base OR sigma protocol is then repeatedly executed defining the main OR sigma protocol to improve soundness. Finally, a Fiat-Shamir transformation turns the sigma protocol into a signature scheme.

Here we will present the base OR sigma protocol with tag, without any of the optimizations of [BKP20], which will be sufficient to highlight the problems of the LIGA instantiation. Furthermore, the A_i's in the following protocol should be sent using a commitment scheme to ensure zero-knowledge, but we have not included this security detail so as not to clutter the presented protocol.

Protocol 1. Σ-protocol $((\mathcal{P}_1, \mathcal{P}_2), (\mathcal{V}_1, \mathcal{V}_2))$ for the relation between the statement $(X_1, \ldots, X_N, T)$ and the witness (s_I, I) such that $X_I = s_I \star X_0$ and $T = s_I \bullet T_0$, illustrated in figure 2.

- **Round 1.** $\mathcal{P}_1((X_1, \ldots, X_N, T), (s_I, I))$
  The prover computes $\mathcal{A} = \{A_i := a \star X_i \mid i \in [\![1, N]\!]\}$ for $a \leftarrow \mathcal{D}_G$ and sends the set to the verifier, with the elements in a random order, and computes $T' = a \bullet T$ and sends it as well.
- **Round 2.** $\mathcal{V}_1((X_1, \ldots, X_N, T), (\mathcal{A}, T'))$
  The verifier responds with a random $c \leftarrow \{0, 1\}$.
- **Round 3.** $\mathcal{P}_2((X_1, \ldots, X_N, T), (s_I, I), c)$
  - If $c = 0$, the prover sends $z = a \cdot s_I$.
  - If $c = 1$, the prover sends $z = a$.
- **Round 4.** $\mathcal{V}_2((X_1, \ldots, X_N, T), (\mathcal{A}, T'), z)$
  - If $c = 0$, the verifier checks that $z \star X_0 \in \mathcal{A}$ and $z \bullet T_0 = T'$.
  - If $c = 1$, the verifier checks that $\{z \star X_i \mid i \in [\![1, N]\!]\} = \mathcal{A}$ and $z \bullet T = T'$.

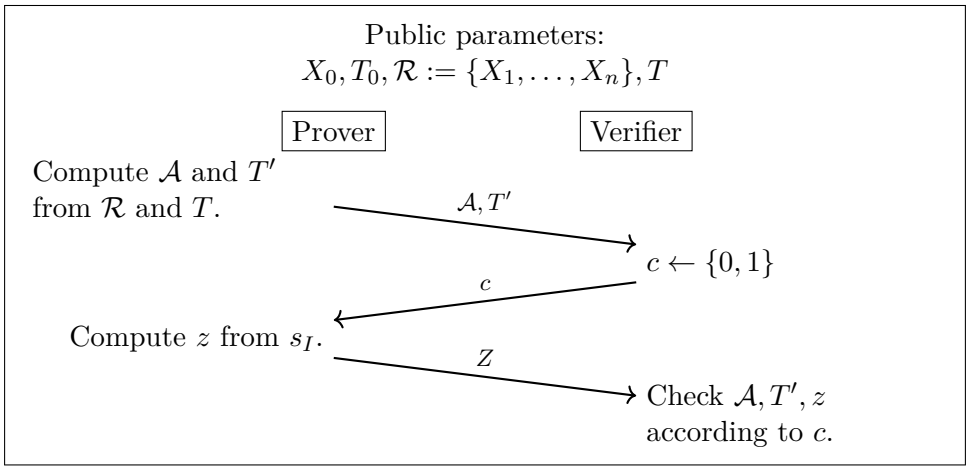

Figure 2: Protocol 1

In plain terms, the prover proves to the verifier their knowledge of a secret key s corresponding both to some public key in the ring $\mathcal{R} = (X_1, \ldots, X_N)$ and to the tag T, without revealing which public key in the ring the tag is associated to. Notice that correctness is verified, for $c = 0$ by

$$z \star X_0 = (a \cdot s_I) \star X_0 = a \star (s_I \star X_0) = a \star X_I = A_I \in \mathcal{A} \tag{1}$$

$$z \bullet T_0 = (a \cdot s_I) \bullet T_0 = a \bullet (s_I \bullet T_0) = a \bullet T = T', \tag{2}$$

and for $c = 1$ by

$$\{z \star X_i \mid i \in [\![1, N]\!]\} = \{a \star X_i \mid i \in [\![1, N]\!]\} = \{A_i \mid i \in [\![1, N]\!]\} = \mathcal{A}$$

$$z \bullet T = a \bullet T = T'.$$

3.2 LIGA instantiation

To turn the general signature scheme construction into an actual LRS, we need to instantiate the two group actions $\star$ and $\bullet$. To do so, in [KTS+24] the authors use LIGA as defined in 2.3.3 to instantiate $\star$, and instantiate $\bullet$ as the inverse of $\star$, as seen here:

$$\bullet : \begin{array}{ccc} \mathrm{GL}_n(\mathbb{Z}) \times [\mathbf{Q}_0] & \to & [\mathbf{Q}_0] \\ (\mathbf{U}, \mathbf{Q}) & \mapsto & \mathbf{U} \bullet \mathbf{Q} := \mathbf{U}^{-1} \star \mathbf{Q} = \mathbf{U}^{-\intercal} \mathbf{Q} \mathbf{U}^{-1} \end{array}$$

This means that we take $G = \mathrm{GL}_n(\mathbb{Z})$ and $\mathcal{X} = \mathcal{T} = [\mathbf{Q}_0]$ for some $\mathbf{Q}_0 \in \mathcal{S}_n^{>0}(\mathbb{Z})$. In [KTS+24], the authors choose to define $\mathbf{Q}_0 := \mathbf{I}_n$, which we will come to later. For X_0 and T_0, they define them as $X_0 = T_0 = \mathbf{Q}_0$, and naturally $\mathsf{Link}_{\mathsf{GA}}(\mathbf{T}, \mathbf{T}') = 1$ when $\mathbf{T} = \mathbf{T}'$, and 0 otherwise. For the distribution $\mathcal{D}_G$, they use the distribution $\mathcal{D}_{\mathrm{GL}_n(\mathbb{Z})}$ described in section 2.3.3 from [DW22, Definition 3.3]. In [KTS+24], the authors assert that this defines an admissible pair of group actions, and therefore benefits from the proof in [BKP20] that the resulting scheme is in fact a linkable ring signature. We will call the signature resulting from this LIGA construction the Lattice Isomorphism Group Action Linkable Ring Signature, *LIGA-LRS*.

We must now bring the reader's attention to the choice of group used in this instantiation, the general linear group $\mathrm{GL}_n(\mathbb{Z})$. It is not a commutative group, and therefore does not strictly fit the framework of [BKP20]. It is then natural to ask whether LIGA-LRS does in fact still benefit from the results proven for commutative groups. We show in the next section that it does not.

4 LIGA-LRS is not a Linkable Ring Signature

As said before, in this section, we will explain the consequence of the non-commutativity of $\mathrm{GL}_n(\mathbb{Z})$ in the context of its usage in LIGA-LRS. In other words, we will explain why the correctness of LIGA-LRS is not provided, and we will present an attack we found against the linkability property of LIGA-LRS, which also relies on the non-commutativity of $\mathrm{GL}_n(\mathbb{Z})$. For the attack against linkable anonymity, we refer the reader to the already mentioned article by Budroni, Chi-Domínguez, and Franch in [BCF25].

4.1 LIGA-LRS is not Correct

Since LIGA-LRS uses the non-commutative group $\mathrm{GL}_n(\mathbb{Z})$, we may consider the handedness of the two actions $\star$ and $\bullet$. It can be seen to see that $\star$ is a right action. Since group inversion is anti-homomorphic, it flips the handedness of the action it is composed with, giving us that $\bullet$ is a left action.

Consider now the equations (1) and (2). For the right action we need $\mathbf{Z} = \mathbf{S}_I \cdot \mathbf{A}$, and for the left action we need $\mathbf{Z} = \mathbf{A} \cdot \mathbf{S}_I$. Because the group is non-commutative, $\mathbf{S}_I \cdot \mathbf{A} = \mathbf{A} \cdot \mathbf{S}_I$ is in general not true, and therefore both equations aren't true simultaneously. This results in the verifier systematically rejecting the proof of knowledge in round 4 of protocol 1, except for the

very rare cases of commutativity. This results in the sigma protocol always rejecting when $c = 0$. Since the sigma protocol is repeated multiple times in the signature to improve the soundness error, the likelihood of one instance of the sigma protocol in the signature happening with $c = 0$ is almost certain.[2] The LRS built from this protocol then accepts signatures from a valid signer with probability nearly 0, and therefore does not fulfill the correctness condition. It was precisely this that made us realize there was something awry. When attempting to implement the signature, we noticed that Verify rejected all the signatures produced, which led us to dig deeper into the correctness of the signature, through the analysis detailed above.

Notice how in this analysis, we have not used any underlying structure of LIGA-LRS other than the fact that the two actions $\star$ and $\bullet$ have opposite handedness. Indeed, for all signatures based on the [BKP20] framework, mixed-handedness of their group actions prevents correctness. This is the case, for example, in the Linear Code Equivalence problem based proposal in [BBN+22]. In that example, the acting group Mono_n is non-commutative, and the actions are defined such that $\star$ is a right action and $\bullet$ is a left action, which leads to this signature not having correctness either. It is not the case, however, for the signature proposed in [DG22]. Despite using the non-commutative group $\mathrm{GL}_n(\mathbb{Z}/q\mathbb{Z})$, their signature still maintains correctness by using two right actions.

4.2 LIGA-LRS is not Linkable

To break the linkability of the ring signature, we need to break the linkability of the group actions $\star$ and $\bullet$. This means finding $\mathbf{U}, \mathbf{V} \in \mathrm{GL}_n(\mathbb{Z})$ such that their public keys $\mathbf{Q} = \mathbf{U} \star \mathbf{Q}_0$ and $\mathbf{Q}' = \mathbf{V} \star \mathbf{Q}_0$ are equal, but their tags $\mathbf{T} = \mathbf{U} \bullet \mathbf{Q}_0$ and $\mathbf{T}' = \mathbf{V} \bullet \mathbf{Q}_0$ are different. We therefore must have

$$
\begin{aligned}
&\mathbf{Q} = \mathbf{Q}' && && \mathbf{T} \neq \mathbf{T}' \\
\Leftrightarrow &\mathbf{U} \star \mathbf{Q}_0 = \mathbf{V} \star \mathbf{Q}_0 && && \Leftrightarrow \mathbf{U} \bullet \mathbf{Q}_0 \neq \mathbf{V} \bullet \mathbf{Q}_0 \\
\Leftrightarrow &\mathbf{U}\mathbf{V}^{-1} \star \mathbf{Q}_0 = \mathbf{Q}_0 && \text{and} && \Leftrightarrow \mathbf{U}^{-1} \star \mathbf{Q}_0 \neq \mathbf{V}^{-1} \star \mathbf{Q}_0 \\
\Leftrightarrow &\mathbf{U}\mathbf{V}^{-1} \in \mathrm{Stab}_\star(\mathbf{Q}_0) && && \Leftrightarrow \mathbf{U}^{-1}\mathbf{V} \star \mathbf{Q}_0 \neq \mathbf{Q}_0 \\
\Leftrightarrow &\mathbf{U}\mathbf{V}^{-1} \in \mathrm{Aut}(\mathbf{Q}_0) && && \Leftrightarrow \mathbf{U}^{-1}\mathbf{V} \notin \mathrm{Stab}_\star(\mathbf{Q}_0) \\
\Leftrightarrow &\mathbf{V}\mathbf{U}^{-1} \in \mathrm{Aut}(\mathbf{Q}_0) && && \Leftrightarrow \mathbf{U}^{-1}\mathbf{V} \notin \mathrm{Aut}(\mathbf{Q}_0).
\end{aligned}
$$

Then all we need to break linkability are two matrices $\mathbf{U}^{-1}, \mathbf{V} \in \mathrm{GL}_n(\mathbb{Z})$ such that their product in one order is a lattice automorphism of $\mathbf{Q}_0$, and the product in the opposite order is not. If the group is commutative, this is impossible, since then $\mathbf{V}\mathbf{U}^{-1} = \mathbf{U}^{-1}\mathbf{V}$.

If the only automorphisms of $\mathbf{Q}_0$ are $\pm\mathbf{I}_n$ then it is again impossible to find such $\mathbf{U}^{-1}$ and $\mathbf{V}$, since then $\mathbf{U} = \pm\mathbf{V}$ and so they must commute. However, in [KTS+24, Section 2.7 Parameter Selection], the authors chose to use $\mathbf{Q}_0 := \mathbf{I}_n$, whose automorphisms are known to be the orthogonal matrices $\mathcal{O}_n(\mathbb{Z})$, many more than just $\pm\mathbf{I}_n$. For some $\mathbf{A} \in \mathcal{O}_n(\mathbb{Z}) \setminus \{\pm\mathbf{I}_n\}$, we wish to find $\mathbf{U}^{-1}, \mathbf{V} \in \mathrm{GL}_n(\mathbb{Z})$ such that $\mathbf{V}\mathbf{U}^{-1} = \mathbf{A}$ and $\mathbf{U}^{-1}\mathbf{V} \notin \mathcal{O}_n(\mathbb{Z})$. This can be reformulated as, for some $\mathbf{A} \in \mathcal{O}_n(\mathbb{Z}) \setminus \{\pm\mathbf{I}_n\}$, we wish to find $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ such that $\mathbf{U}^{-1}\mathbf{A}\mathbf{U} \notin \mathcal{O}_n(\mathbb{Z})$. We now formulate the following ad-hoc assumption.

Assumption 1. *Let* $\mathbf{A} \xleftarrow{\$} \mathcal{O}_n(\mathbb{Z}) \setminus \{\pm\mathbf{I}_n\}$, *and* $\mathbf{U} \leftarrow \mathcal{D}_{\mathrm{GL}_n(\mathbb{Z})}$. *Then* $\mathbf{U}^{-1}\mathbf{A}\mathbf{U} \notin \mathcal{O}_n(\mathbb{Z})$ *with high probability.*

We again wrote Sage code to experimentally validate this assumption. We found that for every one of 100 trials for values of n ranging from 10 to 50, $\mathbf{A} \in \mathcal{O}_n(\mathbb{Z})$ but $\mathbf{U}^{-1}\mathbf{A}\mathbf{U} \notin \mathcal{O}_n(\mathbb{Z})$, validating the assumption. This results in the linkability of LIGA-LRS being broken when $\mathbf{Q}_0 = \mathbf{I}_n$.

[2] If we follow the exact description in [BKP20] of the signature construction, the number of challenges equal to 0 or 1 is fixed, so getting such a $c = 0$ is actually certain.

5 Attempting to correct the Signature

We have seen that LIGA-LRS lacks *correctness* and *linkability*, and by [BCF25] it also lacks *linkable anonymity*, 3 of the properties required to be a linkable ring signature. Most importantly, it lacks correctness due to $\star$ being a right action while $\bullet$ is a left action. In order to fix this, we may on one hand consider alternative ways of defining $\bullet$ so that it also is a right action, and on the other hand, we may consider restricting the secret key space to a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$, so that the handedness of both actions isn't at play.

5.1 Actions of the form $\mathbf{U} \bullet \mathbf{Q} := f(\mathbf{U}) \star \mathbf{Q}$

From the LIP problem, we showed how to construct a one-way group action $\star$. For linkable ring signatures following the [BKP20] framework, we also need a second action $\bullet$. Furthermore, as shown in section 4.1, if we want correctness, both actions must have the same handedness. It is quite natural to want to describe $\bullet$ in terms of $\star$, i.e. $\mathbf{U} \bullet \mathbf{Q} := f(\mathbf{U}) \star \mathbf{Q}$ for some f, so as to potentially benefit from the cryptographic properties of $\star$. In LIGA-LRS, f is chosen to be $f : x \mapsto x^{-1}$, but given that it is anti-homomorphic, $\star$ and $\bullet$ have opposite handedness. However, if we construct $\bullet$ from $\star$ and a group endomorphism f of $\mathrm{GL}_n(\mathbb{Z})$, we ensure that $\bullet$ and $\star$ have the same handedness, thus ensuring correctness.

5.1.1 Necessary condition for Linkable Anonymity

Consider the case of f an automorphism of $\mathbb{R}^{n\times n}$, or of any subgroup of $\mathbb{R}^{n\times n}$ containing $\mathrm{GL}_n(\mathbb{Z})$ and $\mathbf{Q}_0 \in \mathcal{S}_n^{>0}(\mathbb{R})$, such that f and f^{-1} can be computed by a PPT algorithm and such that $\forall \mathbf{U} \in \mathrm{GL}_n(\mathbb{Z}), f(\mathbf{U}^\intercal) = f(\mathbf{U})^\intercal$. We have a public key $\mathbf{X} = \mathbf{U} \star \mathbf{Q}_0$ and a tag $\mathbf{T} = \mathbf{U} \bullet \mathbf{Q}_0 = f(\mathbf{U}) \star \mathbf{Q}_0$. From this, we may compute

$$\begin{aligned} f^{-1}(\mathbf{T}) &= f^{-1}(f(\mathbf{U}) \star \mathbf{Q}_0) \\ &= f^{-1}(f(\mathbf{U})^\intercal \mathbf{Q}_0 f(\mathbf{U})) \\ &= f^{-1}(f(\mathbf{U}^\intercal) \mathbf{Q}_0 f(\mathbf{U})) \\ &= \mathbf{U}^\intercal f^{-1}(\mathbf{Q}_0)\mathbf{U} \\ &= \mathbf{U} \star f^{-1}(\mathbf{Q}_0). \end{aligned}$$

We therefore have a pair of LIP instances $(\mathbf{Q}_0, \mathbf{X})$ and $(f^{-1}(\mathbf{Q}_0), f^{-1}(\mathbf{T}))$ with the same secret $\mathbf{U}$, which we may distinguish from random as shown in [BCF25]. Thus, any pair of group actions constructed as such does not have linkable anonymity.

This is true for example for any f taken from $\mathrm{Aut}(\mathrm{GL}_n(\mathbb{Z}))$[3] with $n > 2$, when $\mathbf{Q}_0$ is invertible over $\mathbb{R}$. It does not apply to $\tau : \mathbf{U} \mapsto \mathbf{U}^{-\intercal} \in \mathrm{Aut}(\mathrm{GL}_n(\mathbb{Z}))$ when $\mathbf{Q}_0$ is not invertible since then τ cannot be extended to a morphism defined over $\mathbf{Q}_0$, but that case can be handled in a more full-fledged Kronecker product argument, analogous to the one in [BCF25, Section 3.1].

5.1.2 Sufficient condition for Linkability

Consider the case of f an endomorphism of $\mathrm{GL}_n(\mathbb{Z})$ that maps automorphisms of $\mathbf{Q}_0$ to automorphisms of $\mathbf{Q}_0$, i.e., $f(\mathrm{Aut}(\mathbf{Q}_0)) \subseteq \mathrm{Aut}(\mathbf{Q}_0)$. Suppose we have two different secret keys $\mathbf{U}$ and $\mathbf{V}$ that produce the same public key $\mathbf{Q} = \mathbf{U} \star \mathbf{Q}_0 = \mathbf{V} \star \mathbf{Q}_0$, and denote $\mathbf{T} = \mathbf{U} \bullet \mathbf{Q}_0$ and

[3] For more details on this group see [HR51, Theorem 4].

$\mathbf{T}' = \mathbf{V} \bullet \mathbf{Q}_0$ the tags associated to these secret keys. Then

$$\begin{aligned}
& \mathbf{U} \star \mathbf{Q}_0 = \mathbf{V} \star \mathbf{Q}_0 \\
\Rightarrow\ & \mathbf{U}\mathbf{V}^{-1} \star \mathbf{Q}_0 = \mathbf{Q}_0 \\
\Rightarrow\ & \mathbf{U}\mathbf{V}^{-1} \in \mathrm{Aut}(\mathbf{Q}_0) \\
\Rightarrow\ & f(\mathbf{U}\mathbf{V}^{-1}) \in \mathrm{Aut}(\mathbf{Q}_0) \\
\Rightarrow\ & f(\mathbf{U}\mathbf{V}^{-1}) \star \mathbf{Q}_0 = \mathbf{Q}_0 \\
\Rightarrow\ & f(\mathbf{U})f(\mathbf{V})^{-1} \star \mathbf{Q}_0 = \mathbf{Q}_0 \\
\Rightarrow\ & f(\mathbf{U}) \star \mathbf{Q}_0 = f(\mathbf{V}) \star \mathbf{Q}_0 \\
\Rightarrow\ & \mathbf{U} \bullet \mathbf{Q}_0 = \mathbf{V} \bullet \mathbf{Q}_0 \\
\Rightarrow\ & \mathbf{T} = \mathbf{T}'.
\end{aligned}$$

It is therefore impossible to produce secret keys $\mathbf{U}, \mathbf{V}$ such that their public keys are equal, but their tags are not. Thus, this construction ensures linkability.

We see then that to define $\bullet$, we want f an $\mathrm{Aut}(\mathbf{Q}_0)$-preserving endomorphism of $\mathrm{GL}_n(\mathbb{Z})$ to ensure linkability, but we also want f to not be an automorphism of $\mathrm{GL}_n(\mathbb{Z})$ to avoid [BCF25]-style attacks on linkable anonymity. Candidates such as $f : x \mapsto x^{-1}$ or $f : x \mapsto x^2$ come to mind, but they're not even endomorphisms, due to the non-commutativity of $\mathrm{GL}_n(\mathbb{Z})$. In fact, other than the trivial constant endomorphism $f : x \to \mathbf{I}_n$ which is unsuitable for cryptographic use, we were not able to come up with a single endomorphism of $\mathrm{GL}_n(\mathbb{Z})$ that was not also an automorphism. Given that finding group actions such that the correctness, linkable anonymity, and linkability properties of the signature hold seems difficult due to the non-commutativity of $\mathrm{GL}_n(\mathbb{Z})$, one may wish to use a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$ instead.

5.2 Commutative subgroups of $\mathrm{GL}_n(\mathbb{Z})$

We now consider restricting the acting group to a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$, in hopes that this will yield more candidate non-automorphic endomorphisms to construct $\bullet$. Unfortunately, we found that the following proposals for commutative subgroups were either impractical or they resulted in a significant decrease in the difficulty of LIP.

5.2.1 Groups generated by a single matrix

The most direct approach to finding commutative subgroups of $\mathrm{GL}_n(\mathbb{Z})$ consists of constructing the commutative subgroup $G := \langle \mathbf{M} \rangle$ of the powers of some matrix $\mathbf{M} \in \mathrm{GL}_n(\mathbb{Z})$ of sufficiently high order. Since this group is meant to represent the secret keys of our signature, we need 2^λ different powers to achieve λ-bit security. We then define the space of secret keys as $\{\mathbf{M}^e \mid e \in [\![0, 2^\lambda - 1]\!]\} \subseteq G$. However, if the coefficients of $\mathbf{M}$ are of size h, then the coefficients of $\mathbf{M}^{2^\lambda}$ are of size roughly $2^\lambda h$. Thus, the size of the secret keys grows exponentially in λ and can't be practically manipulated in polynomial time, nor stored in polynomial space.

To manage the problem of exponential-size coefficients, we consider the subgroup $G := \langle \mathbf{M}_1, \ldots, \mathbf{M}_m \rangle$ generated by multiple matrices of $\mathrm{GL}_n(\mathbb{Z})$ of sufficiently high order that commute with one another. We may then consider the space of secret keys to be $\{\prod_{i=1}^m \mathbf{M}_i^{e_i} \mid e_i \in [\![0, 2^{\lambda/m} - 1]\!]\} \subseteq G$. We then have up to 2^λ possible secret keys, and if the coefficients of the $\mathbf{M}_i$'s are of size h, then the coefficients of the secret keys are of size $2^{\lambda/m}mh$. In particular, if we have $m = \lambda$ different commuting matrices, then the size of coefficients of the secret keys is $2\lambda h$, polynomial in λ. All that remains is to find such commuting $\mathbf{M}_1, \ldots, \mathbf{M}_\lambda$.

Note that this construction may be practical even with less than λ matrices. For example, for $\lambda = 128$ bits of security with $m = 16$ matrices whose entries are of size $h = 4$ bits, we get secret key entries of size $\approx 2^{\lambda/m}mh = 16384$ bits. Considering n scales roughly like λ, the secret keys have roughly λ^2 entries of roughly 16384 bits, resulting in a (relatively) reasonable secret key size of 32MB. Be that as it may, we found that drawing just 3 random matrices of small dimension ($n < 10$) that are of sufficient order and that all commute with each other to be far from trivial.

5.2.2 Shear matrices

One family of commutative matrices in $\mathrm{GL}_n(\mathbb{Z})$ is the family of shear matrices, that is matrices of the form

$$\mathbf{S} := \begin{bmatrix} 1 & s_1 & s_2 & \cdots & s_{n-1} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Not only do shear matrices commute with one another, but they also form a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$. This means that for public shears $\mathbf{S}_1, \dots \mathbf{S}_\lambda$, we may work over the group $G := \langle \mathbf{S}_1, \dots, \mathbf{S}_\lambda \rangle$, and we have that a secret key $\mathbf{U} \in \{\prod_{i=1}^{\lambda} \mathbf{S}_i^{e_i} \mid e_i \in \{0,1\}\}$ is also a shear and therefore of the form

$$\mathbf{U} := \begin{bmatrix} 1 & u_1 & u_2 & \cdots & u_{n-1} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

We now consider the LIP problem in which we know the secret isomorphism is a shear matrix. Given a LIP instance $(\mathbf{Q}_0, \mathbf{Q} := \mathbf{U}^\intercal \mathbf{Q}_0 \mathbf{U})$ with $\mathbf{U}$ of this form, is it still difficult to find $\mathbf{U}$? Considering that $\mathbf{U}$ is defined by $n-1$ coefficients, yet the matrix equation $\mathbf{Q} = \mathbf{U}^\intercal \mathbf{Q}_0 \mathbf{U}$ yields $\frac{n(n+1)}{2}$ equations (not n^2, due to the symmetry of the matrix), we may suspect that there is a simple method for computing the coefficients of $\mathbf{U}$ from those of $\mathbf{Q}_0$ and $\mathbf{Q}$. If we write the coefficients of $\mathbf{Q}_0$ and $\mathbf{Q}$ as $(q_{ij})_{i,j \in [\![0,n-1]\!]}$ and $(q'_{ij})_{i,j \in [\![0,n-1]\!]}$, we get that $\forall j \in [\![1, n-1]\!], q'_{0j} = q_{00} u_j + q_{0j}$, and thus finding $\mathbf{U}$ from $(\mathbf{Q}_0, \mathbf{Q})$ can be done in polynomial time, except for when $q_{00} = 0$. In that case, we instead have that $\forall j \in [\![1, n-1]\!], q'_{jj} = 2 q_{0j} u_j + q_{jj}$, and we may still find $\mathbf{U}$. LIP restricted to shear matrices is therefore not hard, and not a viable solution.

We may instead consider working over $G := \mathbf{P} \langle \mathbf{S}_1, \dots, \mathbf{S}_\lambda \rangle \mathbf{P}^{-1}$ for some public $\mathbf{P} \in \mathrm{GL}_n(\mathbb{Z})$. This subgroup of $\mathrm{GL}_n(\mathbb{Z})$ is still commutative, but its matrices are no longer shears. This however does not mitigate the attack on LIP, since for a LIP instance $(\mathbf{Q}_0, \mathbf{Q})$ with a secret from this group, $(\mathbf{P}^\intercal \mathbf{Q}_0 \mathbf{P}, \mathbf{P}^\intercal \mathbf{Q} \mathbf{P})$ is a LIP instance with a secret from the group of shears, and thus no harder than what we had before introducing $\mathbf{P}$.

5.2.3 Polynomials of a nilpotent matrix

We now consider a different approach for obtaining a commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$, using *nilpotent* matrices. To do so, we remind the reader of the definition and some properties of nilpotent matrices.

Definition 11 (Nilpotent Matrix)**.** *A square matrix $\mathbf{N} \in \mathbb{R}^{n \times n}$ is said to be* nilpotent *if there exists $k \in \mathbb{N}$ such that $\mathbf{N}^k = \mathbf{0}_n$. The smallest such k is called the* index of nilpotency *of $\mathbf{N}$.*

Properties 1.

1. *For any nilpotent matrices $\mathbf{A}$ and $\mathbf{B}$ that commute and any $c \in \mathbb{R}$, $\mathbf{A} + \mathbf{B}$, $\mathbf{AB}$, and $c\mathbf{A}$ are nilpotent. In particular, this implies that given a nilpotent $\mathbf{N} \in \mathbb{R}^{n \times n}$, all no-constant-term polynomials in $\mathbf{N}$ are nilpotent matrices as well. More precisely, all elements of the following set are nilpotent: $\{P(\mathbf{N}) \mid P \in \mathbb{R}[\mathbf{X}], P(\mathbf{0}_n) = \mathbf{0}_n\}$.*

2. *The only eigenvalue of a nilpotent matrix is 0 (can be deduced from [HJ12, Theorem 1.1.6]).*

3. *For any nilpotent matrix $\mathbf{N}$, we have that $\det(\mathbf{I}_n + \mathbf{N}) = 1$ (can be deduced from the previous point and from [HJ12, Theorem 1.1.6]).*

4. *The index of nilpotency of a nilpotent matrix is bounded by its dimension.*

Now, consider a *non-zero* nilpotent matrix $\mathbf{N} \in \mathbb{Z}^{n\times n}$ with index of nilpotency k. We define our group as the polynomials of $\mathbf{N}$ of constant term $\mathbf{I}_n$ and degree less than k under matrix multiplication:

$$G := \left\{ \mathbf{I}_n + \sum_{i=1}^{k-1} p_i \mathbf{N}^i \;\middle|\; p_1, p_2, \ldots, p_{k-1} \in \mathbb{Z} \right\} \subset \mathrm{GL}_n(\mathbb{Z}).$$

Proposition 1. *G is an infinite commutative subgroup of $\mathrm{GL}_n(\mathbb{Z})$ in bijection with $\mathbb{Z}^{k-1}$, with the bijection given by*

$$\begin{array}{rccc} \phi : & \mathbb{Z}^{k-1} & \to & G \\ & (u_1, \ldots, u_{k-1}) & \mapsto & \mathbf{I}_n + \sum_{i=1}^{k-1} u_i \mathbf{N}^i \end{array}$$

giving us a natural way to uniquely represent the elements of G.

Proof. G verifies the following properties.

- *Inclusion.* Any $\mathbf{U} \in G$ has integer entries, and by properties 1. and 3. from above, we also have that $\det \mathbf{U} = 1$, so $G \subseteq \mathrm{GL}_n(\mathbb{Z})$.
- *Identity element.* The identity element is $\mathbf{I}_n \in G$.
- *Closure under multiplication.* For any $\mathbf{U}, \mathbf{U}' \in G$, we have that $\mathbf{U} = P(\mathbf{N}), \mathbf{U}' = P'(\mathbf{N})$ where $P, P' \in \mathbb{Z}[\mathbf{X}]$ with $\deg P, \deg P' \leq k-1$ and $P(\mathbf{0}_n) = P'(\mathbf{0}_n) = \mathbf{I}_n$. Then $\mathbf{U}\mathbf{U}' = P(\mathbf{N})P'(\mathbf{N}) = PP'(\mathbf{N})$ with $PP'(\mathbf{0}_n) = P(\mathbf{0}_n)P'(\mathbf{0}_n) = \mathbf{I}_n$. Since the terms of PP' containing $\mathbf{N}^m$ with $m \geq k$ are zero due to the nilpotency of $\mathbf{N}$, $PP'(\mathbf{N})$ is of degree less than k, and thus $\mathbf{U}\mathbf{U}' \in G$.
- *Commutativity.* For any $\mathbf{U}, \mathbf{U}' \in G$, we have $\mathbf{U}\mathbf{U}' = \mathbf{U}'\mathbf{U}$ since they are polynomials of the same matrix $\mathbf{N}$.
- *Inverse.* Let $\mathbf{U} = \mathbf{I}_n + \sum_{i=1}^{k-1} u_i \mathbf{N}^i \in G$, and let's try finding its inverse $\mathbf{V} = \mathbf{I}_n + \sum_{i=1}^{k-1} u_i \mathbf{N}^i \in G$. We set the equation $\mathbf{U}\mathbf{V} = \mathbf{I}_n$ and we develop.

$$\begin{aligned} &\left(\mathbf{I}_n + \sum_{i=1}^{k-1} u_i \mathbf{N}^i\right)\left(\mathbf{I}_n + \sum_{i=1}^{k-1} v_i \mathbf{N}^i\right) = \mathbf{I}_n \\ \Rightarrow\ &\mathbf{I}_n + \sum_{i=1}^{k-1} \left(u_i + v_i + \sum_{j+k=i} u_j v_k\right) \mathbf{N}^i = \mathbf{I}_n \end{aligned}$$

  Now, it suffices to notice that we can find integers $v_1, \ldots, v_{d-1}$ that satisfy the above equation. To achieve this, we need $u_i + v_i + \sum_{j+k=i} u_j v_k = 0$ for all i. For $i = 1$, we can simply take $v_1 := -u_1$ (the sum is empty for $i = 1$). For $i = 2$, we plug in the value of v_1 we just found and we solve for v_2, then for $i = 3$ we plug in the value of v_2 and we solve for v_3, etc., all the way up to $i = d-1$. Notice that because of the specific form of these equations, we only use addition and multiplication while solving for the unknown v_i's, thus we are guaranteed to find an integer solution, which defines the inverse of $\mathbf{U}$, $\mathbf{U}^{-1} = \mathbf{V} \in G$.
- *ϕ is bijective.* We prove it by contradiction. Suppose there exist $u = (u_1, \ldots, u_{k-1}), v = (v_1, \ldots, v_{k-1}) \in \mathbb{Z}^{k-1}$ such that $u \neq v$ and $\phi(u) = \phi(v)$. Denote $I := \{i \in [1, k-1] \mid u_i \neq v_i\}$ the set of indices on which the tuples u and v differ. We have $I \neq \emptyset$ since $u \neq v$. We also have that

$$\begin{aligned} &\phi(u) = \phi(v) \\ \Rightarrow\ &\mathbf{I}_n + \textstyle\sum_{i=1}^{k-1} u_i \mathbf{N}^i = \mathbf{I}_n + \sum_{i=1}^{k-1} v_i \mathbf{N}^i \\ \Rightarrow\ &\textstyle\sum_{i\in I} u_i \mathbf{N}^i = \sum_{i\in I} v_i \mathbf{N}^i \\ \Rightarrow\ &\textstyle\sum_{i\in I} (u_i - v_i) \mathbf{N}^i = \mathbf{0}_n \end{aligned}$$

Since $I \neq \emptyset$, we can set $m := \min I$ and factor the left-hand side to obtain

$$\begin{aligned} \mathbf{N}^m \left(\textstyle\sum_{i\in I}(u_i - v_i)\mathbf{N}^{i-m}\right) &= \mathbf{0}_n \\ \Rightarrow \mathbf{N}^m \left((u_m - v_m)\mathbf{I}_n + \textstyle\sum_{i\in I, i\neq m}(u_i - v_i)\mathbf{N}^{i-m}\right) &= \mathbf{0}_n \end{aligned}$$

Denote $\mathbf{M} := \mathbf{I}_n + \sum_{i\in I, i\neq m} \frac{u_i - v_i}{u_m - v_m}\mathbf{N}^{i-m}$, which allows us to rewrite the last equation as $\mathbf{N}^m(u_m - v_m)\mathbf{M} = \mathbf{0}_n$. By the properties 1 of nilpotent matrices, $\mathbf{M} \in \mathrm{GL}_n(\mathbb{Q})$, so $\mathbf{M}$ is invertible, which means that $\mathbf{N}^m = \mathbf{0}_n(u_m - u_v)^{-1}\mathbf{M}^{-1} = \mathbf{0}_n$. However, $\mathbf{N}^m$ can't be zero since m is smaller than the index of nilpotency of $\mathbf{N}$ and since $\mathbf{N}$ is non-zero, obtaining our contradiction.

□

We now consider the LIP problem in which we know the secret isomorphism is taken from the above-defined group G. Let $\mathbf{U} = \mathbf{I}_n + \sum_{i=1}^{k-1} u_i\mathbf{N}^i \in G$, $\mathbf{Q}_0 \in \mathcal{S}_n^{>0}(\mathbb{R})$, and $\mathbf{Q} = \mathbf{U} \star \mathbf{Q}_0 = \mathbf{U}^T\mathbf{Q}_0\mathbf{U}$ be the associated LIP instance. We want to retrieve $\mathbf{U}$ from $\mathbf{Q}$ and $\mathbf{Q}_0$. To do so, we can build a system where the unknowns are the $u_1, \ldots, u_{k-1}$, and the system's coefficients are the coefficients of the powers of the matrix $\mathbf{N}$.

We will now study the number of equations and unknowns of the system. First, since $\mathbf{Q}$ is symmetric, the equation $\mathbf{Q} = \mathbf{U}^T\mathbf{Q}_0\mathbf{U}$ yields $\frac{n(n+1)}{2}$ equations for the system. For the unknowns, we observe that

$$\mathbf{U}^T\mathbf{Q}_0\mathbf{U} = \mathbf{Q}_0 + \sum_{i=1}^{k-1} u_i((\mathbf{N}^i)^\intercal\mathbf{Q}_0 + \mathbf{Q}_0\mathbf{N}^i) + \sum_{1\leq i,j\leq k-1} u_iu_j(\mathbf{N}^i)^\intercal\mathbf{Q}_0\mathbf{N}^i.$$

We can then see that after linearization, in addition to the $k-1$ unknowns represented by the $u_1, \ldots, u_{k-1}$, we also have $\binom{k-1}{2} = \frac{(k-2)(k-1)}{2}$ unknowns represented by the $u_iu_j, 1 \leq i \leq j \leq k-1$, for a total of $\frac{(k-1)k}{2}$ unknowns. We then have a linearized system of $\frac{n(n+1)}{2}$ equations and $\frac{(k-1)k}{2}$ unknowns with $k \leqslant n$. In other words, the number of equations is larger than the number of unknowns, and it is always greater. This suggests that solving LIP over this group is easier than cryptographically required, thus again not a viable solution.

References

[ADM+20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. *Cryptographic Group Actions and Applications*. Cryptology ePrint Archive, Paper 2020/1188. 2020. URL: https://eprint.iacr.org/2020/1188.

[BBC+24] Benjamin Benčina, Alessandro Budroni, Jesús-Javier Chi-Domínguez, and Mukul Kulkarni. *Properties of Lattice Isomorphism as a Cryptographic Group Action*. Cryptology ePrint Archive, Paper 2023/1093. 2024. URL: https://eprint.iacr.org/2023/1093.

[BBD+25] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. *Hawk version 1.1 (February 5, 2025)*. National Institute of Standards and Technology. 2025. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/hawk-spec-round2-web.pdf.

[BBN+22] Alessandro Barenghi, Jean-Francois Biasse, Tran Ngo, Edoardo Persichetti, and Paolo Santini. *Advanced Signature Functionalities from the Code Equivalence Problem*. Cryptology ePrint Archive, Paper 2022/710. 2022. URL: https://eprint.iacr.org/2022/710.

[BCF25] Alessandro Budroni, Jesús-Javier Chi-Domínguez, and Ermes Franch. *Don't Use It Twice: Reloaded! On the Lattice Isomorphism Group Action*. Cryptology ePrint Archive, Paper 2025/516. 2025. DOI: https://doi.org/10.62056/ay76chdj. URL: https://eprint.iacr.org/2025/516.

[BKP20] Ward Beullens, Shuichi Katsumata, and Federico Pintore. *Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices*. Cryptology ePrint Archive, Paper 2020/646. 2020. URL: https://eprint.iacr.org/2020/646.

[BY90] Gilles Brassard and Moti Yung. *One-Way Group Actions*. Springer Berlin Heidelberg. 1990. DOI: 10.1007/3-540-38424-3_7.

[Cou06] Jean-Marc Couveignes. *Hard Homogeneous Spaces*. Cryptology ePrint Archive, Paper 2006/291. 2006. URL: https://eprint.iacr.org/2006/291.

[DG22] Giuseppe D'Alconzo and Andrea Gangemi. *TRIFORS: LINKable Trilinear Forms Ring Signature*. Cryptology ePrint Archive, Paper 2022/1170. 2022. DOI: https://doi.org/10.1142/S0219498826502841. URL: https://eprint.iacr.org/2022/1170.

[DW22] Léo Ducas and Wessel van Woerden. *On the Lattice Isomorphism Problem, Quadratic Forms, Remarkable Lattices, and Cryptography*. Cryptology ePrint Archive, Paper 2021/1332. 2022. DOI: 10.1007/978-3-031-07082-2_23. URL: https://eprint.iacr.org/2021/1332.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design". In: *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. 1986, pp. 174–187. DOI: 10.1109/SFCS.1986.47.

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. 2nd ed. Cambridge University Press, 2012. URL: https://www.anandinstitute.org/pdf/Roger_A.Horn.%20_Matrix_Analysis_2nd_edition(BookSee.org).pdf.

[HR51] Loo-keng Hua and Irving Reiner. "Automorphisms of the unimodular group". In: *Transactions of the American Mathematical Society* 71 (1951), pp. 331–348. URL: https://api.semanticscholar.org/CorpusID:119821210.

[KTS+24] Xuan Thanh Khuc, Anh The Ta, Willy Susilo, Dung Hoang Duong, Fuchun Guo, Kazuhide Fukushima, and Shinsaku Kiyomoto. *Logarithmic-Size (Linkable) Ring Signatures from Lattice Isomorphism Problems*. Security, Privacy, and Applied Cryptography Engineering. 2024. DOI: 10.1007/978-3-031-51583-5_13.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. "How to Leak a Secret". In: *Advances in Cryptology — ASIACRYPT 2001*. Ed. by Colin Boyd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 552–565. ISBN: 978-3-540-45682-7.