Properties of Lattice Isomorphism as a Cryptographic Group Action

Benjamin Benčina[1], Alessandro Budroni[2], Jesús-Javier Chi-Domínguez[2], and Mukul Kulkarni[2]

[1] Royal Holloway, University of London, UK
benjamin.bencina.2022@live.rhul.ac.uk
[2] Cryptography Research Center, Technology Innovation Institute, Abu Dhabi, UAE
{alessandro.budroni,jesus.dominguez,mukul.kulkarni}@tii.ae

Abstract. In recent years, the Lattice Isomorphism Problem (LIP) has served as an underlying assumption to construct quantum-resistant cryptographic primitives, e.g. the zero-knowledge proof and digital signature scheme by Ducas and van Woerden (Eurocrypt 2022), and the HAWK digital signature scheme (Asiacrypt 2022).

While prior lines of work in group action cryptography, e.g. the works of Brassard and Yung (Crypto 1990), and more recently Alamati, De Feo, Montgomery and Patranabis (Asiacrypt 2020), focused on studying the discrete logarithm problem and isogeny-based problems in the group action framework, in recent years this framing has been used for studying the cryptographic properties of computational problems based on the difficulty of determining equivalence between algebraic objects. Examples include Permutation and Linear Code Equivalence Problems used in LESS (Africacrypt 2020), and the Tensor Isomorphism Problem (TCC 2019). This study delves into the quadratic form version of LIP, examining it through the lens of group actions.

In this work we (1) give formal definitions and study the cryptographic properties of this group action (LIGA), (2) demonstrate that LIGA lacks both weak unpredictability and weak pseudorandomness, and (3) under certain assumptions, establish a theoretical trade-off between time complexity and the required number of samples for breaking weak unpredictability, for large dimensions. We also conduct experiments supporting our analysis. Additionally, we employ our findings to formulate new hard problems on quadratic forms.

Keywords: Gröbner Bases · Group Actions · Lattice-based Cryptography · Lattice Isomorphism Problem · Quadratic Forms

1 Introduction

Post-Quantum Cryptography is an active research area which aims to design public-key cryptographic primitives that can resist the threats posed by large scale quantum computers. Since most of the widely used public-key cryptographic algorithms will be affected by the attacks harnessing the computational power of quantum computing, the National Institute of Standards and Technology (NIST), has already selected a few candidates for standardization [40], and more candidates are under consideration [39].

Equivalence problems in cryptography and group actions. The computational hardness of the equivalence problems for algebraic or geometric structures has emerged as an attractive underlying assumption for designing post-quantum cryptographic schemes. Informally, these are search problems which aim to find a map between two equivalent algebraic or geometric objects. Perhaps the most notable example of this approach is isogeny-based cryptography which relies on the hardness of finding isogenies between supersingular elliptic curves [20,19,11,1,23]. Cryptographic schemes have also been designed based on problems related to lattice isomorphisms [25,8], code equivalence [12], trilinear forms [44], and tensor isomorphism [32]. These have shown potential in constructing remarkable primitives, especially in the domains of proofs-of-knowledge and digital signatures [12,24,14,33]. Each of these problems is interesting in its own regard and provides different trade-offs as well as flexibility while designing cryptographic schemes, however these can also be seen as instances of a more general framework based on group actions introduced by [17] and later studied in [20,32,1]. These works show that this class of problems can be modelled as problems related to the computational hardness of inverting a group action.

Lattice isomorphisms as a group action. In this work, we show how to characterize and analyze the quadratic form representation of lattice isomorphisms as a group action (LIGA). We believe that such a characterization helps in unifying the similar computational assumptions under a common framework, which can then be used to study the similarities between these hard problems.

Informally, the *Lattice Isomorphism Problem* (LIP) in its search version aims to find an isomorphism between two given isomorphic lattices. The decision version of the problem asks whether two given lattices are isomorphic or not. Lattice isomorphisms were studied and used initially in the cryptanalysis of early lattice-based schemes such as NTRU [30]. Later, Haviv and Regev studied the complexity of search-LIP [31]. More recently, two independent works by Bennett et al. [8] and by Ducas and van Woerden [25] proposed to use LIP for building cryptographic primitives. Subsequently, a digital signature scheme HAWK based on a module version of LIP has been proposed with impressive results in terms of efficiency and object sizes [24]. In [25,24] the authors focus on the quadratic form representation of the lattice isomorphism problem, which is also the focus of this paper. We find that LIGA is not a weakly unpredictable and therefore also not a weakly pseudorandom group action. These properties are stronger than one-wayness and have been studied in [1] for the isogeny-based group action. Our finding poses a significant barrier to using LIGA as a building block in the constructions of a variety of cryptographic primitives such as the Naor-Reingold-style PRF [38], along with the ones considered in [1, Section 4].

Shortly after the publication of the preprint version of this manuscript [18], another preprint analyzing code equivalence and other problems using a similar approach to our work was published [22].

1.1 Overview of our results

In this paper, we study the Lattice Isomorphism Problem (in the quadratic form setting) as defined in [25], in the framework of group actions with aim to understand its utility in building quantum-secure cryptographic primitives. Our main contributions are summarized below:

Formalizing lattice isomorphisms as a group action. In this work, we formalize LIGA as a group action and prove that it is faithful as well as transitive. We also show that LIGA is free if and only if the automorphism group of the quadratic form Q, defining the underlying equivalence class $[Q]$, is trivial.

Breaking the weak pseudorandomness of LIGA. We generalize the definitions of cryptographic properties of group actions presented in [1] to the setting where the underlying group and set are both infinite.[3] Assuming the conjectured hardness of LIP immediately implies that LIGA is a one-way group action. We then show that LIGA is not a weakly unpredictable (and therefore also not a weakly pseudorandom) group action, under some mild assumptions on the distribution $\mathcal{D}_{[Q]}$ used for sampling challenge instances.

Theorem 1 (Informal). *Given $O(n^2)$ instances of* LIP *obtained from a* fixed *secret isomorphism represented by $n \times n$ unimodular matrix U, it is possible to recover the secret isomorphism U in polynomial time with overwhelming probability.*

We also provide an alternate approach to recover the secret isomorphism U using Gröbner bases. At the cost of an extra multiplicative factor of $O(n^{2+(i-2)\omega})$ to the time complexity of the recovery, for a small constant $i \geq 2$ and some $\omega \in [2, 3]$, one can retrieve the secret by requiring a factor of i^2 fewer samples, while assuming similar mild assumptions on the distribution $\mathcal{D}_{[Q]}$ used for sampling challenge instances.

Introducing two new computationally hard problems. We introduce two new hard problems on quadratic forms: the Transpose Quadratic Form Problem (TQFP) and the Inverse Quadratic Form Problem (IQFP). We use the aforementioned result to demonstrate the equivalence of these problems to search-LIP through dimension-preserving polynomial-time reductions, specifically for quadratic forms with a trivial automorphism group.

It is worth highlighting that such an inverse problem was previously introduced in the context of isogeny-based group actions, namely as Inv-HHS in [27]. In contrast to isogenies, we show that IQFP is as hard as LIP, while in isogenies Inv–HHS is not hard. Due to the non-linear nature of the isogeny-based group action, it appears that solving Inv–HHS does not translate to recovering the secret isogeny in polynomial time.

[3] This is in contrast to most of the prior works which consider group actions related to finite groups and/or sets. In [32] the authors mentioned that their definitions can be used in the setting of infinite groups and sets, specifically aiming at LIP. However, they do not present LIP in the quadratic form setting and only consider the finite groups and sets for their analysis.

Experiments. We verify our results by running experiments for lattice dimensions up to 40. Specifically, we verify that one can indeed recover the secret using estimated number of samples efficiently. We also compare the two approaches and verify that using Gröbner bases allows recovery of the secret from fewer samples. Additionally, we also conduct experiments validating the reductions from search version of LIP to the new problems introduced in this work (TQFP, and to IQFP). All these experiments are performed via a SageMath [45] implementation that is available at [9].

1.2 Organization of the paper

In Section 2 we give the preliminaries on lattices, group actions, and Gröbner basis computation. In Section 3 we formalize lattice isomorphism as a group action and provide the related results. In Section 4 we discuss the Gröbner basis approach along with some experimental results. In Section 5 we introduce two new hard problems together with their reductions from LIP. Finally, in Section 6 we discuss some interesting open problems.

2 Preliminaries

2.1 Notation

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the sets of natural, integer, rational, and real numbers, respectively. We denote vectors in boldface (e.g. $\boldsymbol{v}$) and treat them as columns unless otherwise specified. We denote matrices by uppercase letters (e.g. M), and vectors of matrices by bold uppercase letters (e.g. $\boldsymbol{M}$). Sets are denoted with calligraphic uppercase letters (e.g. $\mathcal{S}$). For a vector $\mathbf{x}$ in $\mathbb{R}^n$, the Euclidean norm is denoted as $\|\mathbf{x}\|$.

The set of all $n \times n$ invertible matrices over $\mathbb{Z}$ is denoted by $\mathrm{GL}_n(\mathbb{Z}) \coloneqq \{M \in \mathbb{Z}^{n\times n} : \det(M) = \pm 1\}$. For an invertible matrix $X \in \mathrm{GL}_n(\mathbb{Z})$, we denote the inverse of the transpose matrix X^{T} as $X^{-\mathsf{T}}$. Also, by I_n we denote the $n \times n$ identity matrix. For a matrix $M = [M_{i,j}] \in \mathbb{Z}^{n\times n}$, denote with $\bar{M}^{(i,j)} \in \mathbb{Z}^{(n-1)\times(n-1)}$ the minor of M with respect to $M_{i,j}$, i.e. the matrix obtained by removing the i-th row and the j-th column from M. We denote by M^* the Gram–Schmidt orthogonalization of M.

A matrix $S \in \mathbb{R}^{n\times n}$ is called *symmetric positive definite* if $S = S^{\mathsf{T}}$ and $\mathbf{x}^{\mathsf{T}} S \mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. The set of all $n \times n$ *symmetric positive definite* matrices over $\mathbb{R}$ is denoted by $\mathcal{S}_n^{>0}$. For $Q = [Q_{i,j}] \in \mathcal{S}_n^{>0}$ and $d \coloneqq \frac{n(n+1)}{2}$, define $\mathsf{unroll}\colon \mathcal{S}_n^{>0} \to \mathbb{R}^d$ as

$$\mathsf{unroll}(Q) \coloneqq \left[Q_{1,1}\ 2Q_{1,2}\ \dots\ 2Q_{1,n}\ Q_{2,2}\ 2Q_{2,3}\ \dots\ 2Q_{2,n}\ \dots\ Q_{n,n}\right].$$

For simplicity, in the remainder of the paper, we assume both matrix multiplication and inversion take $O(n^{\omega})$ integer operations for some $\omega \in [2,3]$.[4]

[4] The Strassen's algorithm is considered to be the best algorithm for large dimensional matrix multiplications with a running time of $O\left(n^{\log_2(7)}\right)$ operations.

Consequently, we assume that solving a linear system $A\mathbf{x} = B$, for some non-singular matrix $A \in \mathbb{Z}^{n\times n}$ and $B \in \mathbb{Z}^{n\times\delta}$ with $\delta \leq n$, takes time $O(n^\omega)$ since it is equivalent to computing $\mathbf{x} = A^{-1}B$.

2.2 Lattice Isomorphisms and Quadratic Forms

We refer the reader to [25] for a detailed introduction on the Lattice Isomorphism Problem. A full-rank n-dimensional lattice $\mathcal{L} = \mathcal{L}(B) \coloneqq B \cdot \mathbb{Z}^n$ is generated by taking all of the possible integer combinations of the columns of a basis $B \in \mathbb{R}^{n\times n}$. Two bases B and B' generate the same lattice if and only if there exists a unimodular matrix $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $B' = BU$.

Let $\mathcal{O}_n(\mathbb{F})$ be the set of all *orthonormal* matrices with entries in a field $\mathbb{F}$. Two lattices $\mathcal{L}$, $\mathcal{L}'$ are *isomorphic* if there exists an orthonormal transformation $O \in \mathcal{O}_n(\mathbb{R})$ such that $\mathcal{L}' = O \cdot \mathcal{L}$.

Definition 1 (Search Lattice Isomorphism Problem (sLIP)). *Given two isomorphic lattices $\mathcal{L}$, $\mathcal{L}' \subset \mathbb{R}^n$ find an orthonormal transformation $O \in \mathcal{O}_n(\mathbb{R})$ such that $\mathcal{L}' = O \cdot \mathcal{L}$.*

The problem in Definition 1 can be rephrased as follows. Given the bases B, $B' \in \mathbb{R}^{n\times n}$ for $\mathcal{L}$ and $\mathcal{L}'$ respectively, find $O \in \mathcal{O}_n(\mathbb{R})$ *along with* $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $B' = OBU$. In practice, the real-valued entries of the bases and orthonormal matrices can be inconvenient to represent and result in inefficient computations. However, this can be eased by considering an equivalent problem to LIP by taking the quadratic form associated to B, i.e. the Gram matrix $Q \coloneqq B^\mathsf{T}B$. Note that the quadratic form Q is symmetric by definition. Moreover, since B is a basis (and thus full-rank), Q is actually *symmetric positive definite*. For isomorphic lattices $\mathcal{L}, \mathcal{L}'$ with respective basis B, B', we have that $B' = OBU$ where $O \in \mathcal{O}_n(\mathbb{R})$ is orthonormal and $U \in \mathrm{GL}_n(\mathbb{Z})$ is unimodular. Then we have

$$Q' \coloneqq {B'}^\mathsf{T}B' = U^\mathsf{T}B^\mathsf{T}O^\mathsf{T}OBU = U^\mathsf{T}B^\mathsf{T}BU = U^\mathsf{T}QU,$$

where $Q \coloneqq B^\mathsf{T}B$ is the quadratic form of B. We call Q, Q' equivalent if such $U \in \mathrm{GL}_n(\mathbb{Z})$ exists. We also denote by $[Q]$ the equivalence class of all quadratic forms Q' equivalent to Q.

Definition 2 (sLIP_Q - Quadratic Form Version). *For a quadratic form $Q \in \mathcal{S}_n^{>0}$, the problem sLIP_Q is, given any quadratic form $Q' \in [Q]$, to find a unimodular $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $Q' = U^\mathsf{T}QU$.*

The squared norm of a vector $\mathbf{x}$ with respect to a quadratic form Q is defined as $\|\mathbf{x}\|_Q^2 \coloneqq \mathbf{x}^\mathsf{T}Q\mathbf{x}$ and the inner product as $\langle \mathbf{x}, \mathbf{y}\rangle_Q \coloneqq \mathbf{x}^\mathsf{T}Q\mathbf{y}$. The i-th minimal distance $\lambda_i(Q)$ is defined as the smallest $r > 0$ such that $\{\boldsymbol{x} \in \mathbb{Z}^n \colon \|\boldsymbol{x}\|_Q \leq r\}$ spans a space of dimension at least i. We denote by B_Q the Cholesky decomposition of Q, that is, an upper triangular matrix such that $Q = {B_Q}^\mathsf{T}B_Q$.

Definition 3 (Automorphisms). *Let $Q \in \mathcal{S}_n^{>0}$ be a quadratic form of dimension n. The automorphism group of Q is defined as*

$$\mathrm{Aut}\,(Q) = \{V \in \mathrm{GL}_n(\mathbb{Z}) \colon Q = V^{\mathsf{T}} Q V\}.$$

We say that Q is automorphism-free if it has a trivial automorphism group $\mathrm{Aut}\,(Q) = \{\pm \mathrm{I}_n\}$.

Remark 1. Let $Q' \in [Q]$, and let $U \in \mathrm{GL}_n(\mathbb{Z})$ be such that $Q' = U^T Q U$. Recall that for any such quadratic form Q, the group $\mathrm{Aut}\,(Q)$ is always finite [42, Section 2.3]. The set of isomorphisms between Q and Q' can be written as $\{VU \colon V \in \mathrm{Aut}(Q)\}$. The automorphism group of Q determines the number of isomorphisms from Q to Q' and vice versa. Therefore equivalent quadratic forms Q and Q' have the same number of automorphisms. Hence, automorphism-free quadratic forms are isomorphic only to automorphism-free quadratic forms. We provide some more details and formal justifications in Section 3 (Corollary 1) when we discuss orbits and stabilizers of lattice isomorphisms when viewed as a group action.

2.3 Sampling Quadratic Forms and Unimodular Matrices

Definition 4 (Discrete Gaussian Distribution w.r.t. Quadratic Forms [25, Sec. 2.3]). *For a quadratic form $Q \in \mathcal{S}_n^{>0}$, the Gaussian function on $\mathbb{R}^n$ with a parameter $s > 0$ and center $\mathbf{c}$ is defined by*

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \rho_{Q,s,\mathbf{c}}(\mathbf{x}) \coloneqq \exp\left(-\pi \|\mathbf{x} - \mathbf{c}\|_Q^2 / s^2\right).$$

The discrete Gaussian distribution $\mathcal{D}_{Q,s,\mathbf{c}}$ is defined as

$$\Pr_{X \sim \mathcal{D}_{Q,s,\mathbf{c}}}[X = \mathbf{x}] \coloneqq \begin{cases} \frac{\rho_{Q,s,\mathbf{c}}(\mathbf{x})}{\rho_{Q,s,\mathbf{c}}(\mathbb{Z}^n)} & \text{if } \mathbf{x} \in \mathbb{Z}^n, \\ 0 & \text{otherwise} \end{cases}.$$

Brakerski et al. [16, Lemma 2.3] showed how to sample from a discrete Gaussian distribution efficiently. Ducas and van Woerden provide a polynomial time algorithm **Extract** that, taking as input a set of n linearly independent vectors Y and a quadratic form Q, returns a pair (Q', U) such that $Q' = U^{\mathsf{T}} Q U$ [25, Lemma 3.1].

Definition 5 (Gaussian Form Distribution [25, Def. 3.3]). *Given a quadratic form equivalence class $[Q] \subset \mathcal{S}_n^{>0}$, the Gaussian form distribution $\mathcal{D}_s\,([Q])$ over $[Q]$ with a parameter $s > 0$ is defined algorithmically as follows:*

1. *Fix a representative $Q \in [Q]$.*
2. *Sample n vectors $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_n) \coloneqq Y$ from $\mathcal{D}_{Q,s}$. Repeat until linearly independent.*
3. *$(R, U) \leftarrow$ **Extract**(Q, Y).*
4. *Return R.*

Ducas and van Woerden also provide a polynomial time algorithm to sample from $\mathcal{D}_s([Q])$ for $s \geq \max\{\lambda_n(Q), \|B_Q^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}$, which returns, together with a quadratic form Q', a unimodular matrix U such that $Q' = U^\mathsf{T} Q U$, and show that $Q' \leftarrow \mathcal{D}_s([Q])$ is independent from the input equivalence class representative Q [25, Lemma 3.2].

Sampling unimodular matrices. The algorithm **Extract** includes a method to derive a unimodular matrix from a set of independent vectors employing the Hermite Normal Form reduction that is folklore in the literature [13,36].

Algorithm 1 is a modified version of [13, Algorithm 4] for sampling unimodular matrices in polynomial time having the entries of the first $n-1$ rows uniform over the integer interval $[-T, T] \subset \mathbb{Z}$, for $T > 0$. For the context of this manuscript, it is not relevant for us whether it produces "cryptographically-strong" random unimodular matrices or not.

2.4 Cryptographic Group Actions

In this section, we present the definitions of cryptographic properties of group actions. Our definitions are inspired by those introduced in [20,32,1]. More specifically, we generalize the prior definitions to hold for infinite groups and sets, and we include the precise number of oracle calls that an adversary is allowed to make for a certain experiment, that is Definition 7 and Definition 8 give more fine-grained notions of weak unpredictability and weak pseudorandomness in comparison to their counterparts in [1, Section 2.1].[5]

Definition 6 (One-Way Functions). *Let P, X and Y be sets drawn from families of sets indexed by the parameter λ, and let $\mathcal{D}_P$ and $\mathcal{D}_X$ be distributions on P and X respectively. A $(\mathcal{D}_P, \mathcal{D}_X)$-OWF family is a family of efficiently computable functions $\{f_{\mathsf{pp}}(\cdot)\colon X \to Y\}_{\mathsf{pp} \in P}$ such that for all PPT adversaries $\mathcal{A}$ we have*

$$\Pr[f_{\mathsf{pp}}(\mathcal{A}(\mathsf{pp}, f_{\mathsf{pp}}(x))) = f_{\mathsf{pp}}(x)] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{pp} \leftarrow \mathcal{D}_P$ and $x \leftarrow \mathcal{D}_X$.

Definition 7 (Weak Unpredictable Permutations). *Let K and X be sets drawn from families of sets indexed by the parameter λ, $\mathcal{D}_K$ and $\mathcal{D}_X$ be distributions on K and X respectively, and $t := t(\lambda) \in \mathbb{N}^+$ be a parameter. Let $F_k^\$$ be a randomized oracle that when queried samples $x \leftarrow \mathcal{D}_X$ and outputs $(x, F(k, x))$. A $(\mathcal{D}_K, \mathcal{D}_X, t)$-weak UP (wUP) is a family of efficiently computable permutations $\{F(k, \cdot)\colon X \to X\}_{k \in K}$ such that for all PPT adversaries $\mathcal{A}$ able to query $F_k^\$$ at most t times, we have*

$$\Pr[\mathcal{A}^{F_k^\$}(x^*) = F(k, x^*)] \leq \mathsf{negl}(\lambda),$$

where $k \leftarrow \mathcal{D}_K$ and $x^* \leftarrow \mathcal{D}_X$.

[5] This fine-grained notion of limiting the number of times the adversary calls the oracle makes our results stronger since any attacker breaking the fine-grained security property also breaks the same property in the sense of [1, Section 2.1] (or other prior definitions of cryptographic properties of group actions.)

Algorithm 1 Sample a unimodular matrix with all rows except the last one having entries uniformly distributed in an integer interval $[-T, T] \subset \mathbb{Z}$.

```
Input: A positive integer parameter $T > 0$.
Output: An $n \times n$ unimodular matrix with all rows except the last one having
    entries uniformly distributed in the integer interval $[-T, T] \subset \mathbb{Z}$.
Set a matrix $M = \{M_{i,j}\} \in \mathbb{Z}^{n \times n}$ to zero.
repeat
    Sample $M_{i,j} \leftarrow [-T, T]$ uniformly at random for each $i \leq n-1$ and
    $j \leq n$.
    Use the Extended Euclidean Algorithm for computing
        $d \leftarrow \gcd\left((-1)^{n+1} \det\left(\bar{M}^{(n,1)}\right), \ldots, (-1)^{2n} \det\left(\bar{M}^{(n,n)}\right)\right),$
    along with the corresponding Bézout coefficients $M_{1,j}$'s such that
        $d = \sum_{j=1}^{n} M_{n,j} \cdot (-1)^{n+j} \det\left(\bar{M}^{(n,j)}\right) = \det(M).$
until $d = 1$.
Choose the sign of $\det(M)$ as follows: sample $b \in \{0, 1\}$ uniformly at random,
swap the first two rows if $b = 1$.
Use least-squares to find the linear combination $\sum_{j=1}^{n-1} c_j [M_{j,1} \ldots M_{j,n}]$ clos-
est to $[M_{n,1} \ldots M_{n,n}]$, and let $\tilde{c}_i$ denote the nearest integer to $c_i$.
Update $[M_{n,1} \ldots M_{n,n}]$ as
    $[M_{n,1} \ldots M_{n,n}] - \sum_{j=1}^{n-1} \tilde{c}_j [M_{j,1} \ldots M_{j,n}].$
Return $M$.
```

Definition 8 (Weak Pseudorandom Permutations). *Let K and X be sets drawn from families of sets indexed by the parameter λ, $\mathcal{D}_K$ and $\mathcal{D}_X$ be distributions on K and X respectively, and $t := t(\lambda) \in \mathbb{N}^+$ be a parameter. Let $\pi^\$$ be a randomized oracle that samples $x \leftarrow \mathcal{D}_X$ and outputs $(x, \pi(x))$, where π is a random permutation on X constructed adaptively by the oracle, i.e. when a new x is queried the oracle $\pi^\$$ samples $y \leftarrow \mathcal{D}_X$ until $y \neq \pi(x')$ for any previously queried $x' \in X$, and sets $\pi(x) := y$. A $(\mathcal{D}_K, \mathcal{D}_X, t)$-weak PRP (wPRP) is a family of efficiently computable permutations $\{F(k, \cdot) \colon X \to X\}_{k \in K}$ such that for all PPT adversaries $\mathcal{A}$ able to query the oracles $F_k^\$$ and $\pi^\$$ at most t times, we have*

$$\left|\Pr[\mathcal{A}^{F_k^\$}(1^\lambda) = 1] - \Pr[\mathcal{A}^{\pi^\$}(1^\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $k \leftarrow \mathcal{D}_K$.[6]

Definition 9 (Group Action). *A group* $(G, \circ)$ *is said to act on a non-empty set* X *if there is a map* $\star\colon G \times X \to X$ *that satisfies the following two properties*

1. *Identity: if* e *is the identity element of* G*, then for any* $x \in X$*, we have* $e \star x = x$.
2. *Compatibility: for any* $g, h \in G$ *and any* $x \in X$*, we have* $(g \circ h) \star x = g \star (h \star x)$.

We use the notation $(G, X, \star)$ *to denote a group action.*

If $(G, X, \star)$ is a group action, for any $g \in G$ the map $\pi_g\colon x \mapsto g \star x$ defines a permutation of X.

Definition 10 (Properties of Group Actions). *A group action* $(G, X, \star)$ *is said to be:*

1. ***transitive****, if for every* $x_1, x_2 \in X$*, there exists a group element* $g \in G$ *such that* $x_2 = g \star x_1$ *(for such a transitive group action, the set* X *is called a homogeneous space for* G*);*
2. ***faithful****, if for each group element* $g \in G$*, either* g *is the identity element or there exists a set element* $x \in X$ *such that* $x \neq g \star x$*;*
3. ***free****, if for every group element* $g \in G$*,* g *is the identity element if and only if there exists some set element* $x \in X$ *such that* $x = g \star x$.

Definition 11 (One-Way Group Action). *A group action* $(G, X, \star)$*, where* G *is a group and* X *is a set indexed by a parameter* λ*, is* $(\mathcal{D}_G, \mathcal{D}_X)$*-one-way if the family of efficiently computable functions* $\{f_x\colon G \to X\}_{x \in X}$ *is* $(\mathcal{D}_G, \mathcal{D}_X)$*-one-way, where* $f_x\colon g \mapsto g \star x$*, and* $\mathcal{D}_G, \mathcal{D}_X$ *are distributions on* G, X *respectively.*

Definition 12 (Weakly Unpredictable Group Action). *A group action* $(G, X, \star)$ *is* $(\mathcal{D}_G, \mathcal{D}_X, t)$*-weakly unpredictable if the family of efficiently computable permutations* $\{\pi_g\colon X \to X\}_{x \in X}$ *is a* $(\mathcal{D}_G, \mathcal{D}_X, t)$*-weak UP, where* π_g *is defined as* $\pi_g\colon x \mapsto g \star x$ *and* $\mathcal{D}_X, \mathcal{D}_G$ *are distributions on* X, G *respectively.*

Definition 13 (Weakly Pseudorandom Group Action). *A group action* $(G, X, \star)$ *is* $(\mathcal{D}_G, \mathcal{D}_X, t)$*-weakly pseudorandom if the family of efficiently computable permutations* $\{\pi_g\colon X \to X\}_{x \in X}$ *is a* $(\mathcal{D}_G, \mathcal{D}_X, t)$*-weak PRP where* π_g *is defined as* $\pi_g\colon x \mapsto g \star x$ *and* $\mathcal{D}_X, \mathcal{D}_G$ *are distributions on* X, G *respectively.*

Orbits and stabilizers of group actions.

Definition 14. *Let* $(G, X, \star)$ *be a group action. The orbit of* $x \in X$ *is a subset of* X *defined as*

$$G \star x = \{g \star x \colon g \in G\},$$

and the stabilizer of $x \in X$ *is the subgroup of* G *defined as*

$$G_x = \{g \in G \colon g \star x = x\}.$$

[6] As the players are PPT, we may model the "random" permutation as a random oracle that uses $\mathcal{D}_X$ to sample new images adaptively.

Remark 2. The orbits of a group action $(G, X, \star)$ partition the set X into disjoint subsets. Moreover, the choice of the orbit representative does not matter, i.e. if $y \in G \star x$ then $G \star x = G \star y$. Note that a group action is transitive if and only if it admits a single orbit.

Remark 3. A well-known fact from group theory is that the stabilizers of set elements from the same orbit are conjugate subgroups of G, i.e. if $x_1 \in G \star x_0$ for $x_0, x_1 \in X$, then there exists $h \in G$ such that $G_{x_1} = hG_{x_0}h^{-1}$. In particular, stabilizers of elements from the same orbit have the same cardinality. Note that a group action is free if and only if all of its stabilizers are trivial. Moreover, the Orbit-Stabilizer Theorem states that for any $x \in X$ the map $f_x \colon G/G_x \to G \star x$ that maps cosets as $hG_x \mapsto h \star x$ is a bijection [35, Theorem 3.2].

2.5 Gröbner Bases and Semi-Regular Sequences

Let $\mathbb{F}$ be a field and $\mathcal{P} = \mathbb{F}[x_1, \ldots, x_n]$ a polynomial ring in n variables over $\mathbb{F}$. Let $\mathcal{M}$ be the set of monomials of $\mathcal{P}$. When $n = 1$ the natural way to order monomials is simply by comparing them by their degree. For $n > 1$, we define orderings that are admissible.

Definition 15. *A* monomial order $<$ *on* $\mathcal{M}$ *(or* $\mathcal{P}$*) is a well-ordering that is compatible with the product on* $\mathcal{M}$*, i.e. for every* $u, v, w \in \mathcal{M}$ *where* $w \geq 1$ *we have that* $u < v$ *implies* $uw < vw$*.*

We identify the set of monomials $\mathcal{M}$ with the set of their coefficients $\mathbb{Z}^n_{\geq 0}$, that is all n-tuples of non-negative integers, where $\alpha = (\alpha_1, \ldots, \alpha_n)$ is identified with the monomial $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. We call α_i the degree of the monomial x at the variable x_i, and we call $|\alpha| = \sum_{i=1}^{n} \alpha_i$ the total degree of x. A monomial order that prioritizes ordering by total degree is called graded.

Definition 16. *The degree reverse lexicographical (DRL) order* $>$ *is defined by* $\alpha > \beta$ *if and only if* $|\alpha| > |\beta|$*, or* $|\alpha| = |\beta|$ *and the last non-zero entry of* $\alpha - \beta$ *is negative.*

Let $\mathcal{P}$ be equipped with the DRL monomial order. For a polynomial $f \in \mathcal{P}$, denote by $\mathrm{LM}(f)$ the leading monomial of f, that is the monomial that is the largest of all monomials of f w.r.t. the monomial order $>$. An ideal $\mathcal{I}$ of $\mathcal{P}$ is called a monomial ideal if there exists a basis for $\mathcal{I}$ consisting of monomials. By Hilbert's Basis Theorem, or more specifically Dickson's Lemma [21, Section 2.4, Theorem 5], such a basis can always be assumed to be finite. For a set $\mathcal{A} \subseteq \mathcal{P}$, we define $\mathrm{LM}(\mathcal{A})$ to be the set of all leading monomials of elements of $\mathcal{A}$.

Definition 17. *Let* $\mathcal{I}$ *be an ideal of* $\mathcal{P}$*. A finite subset* $\mathcal{G} \subset \mathcal{I}$ *is called a* Gröbner basis *of* $\mathcal{I}$ *if the set* $\mathrm{LM}(\mathcal{G})$ *generates the monomial ideal generated by* $\mathrm{LM}(\mathcal{I})$*, i.e.* $\langle \mathrm{LM}(\mathcal{G}) \rangle = \langle \mathrm{LM}(\mathcal{I}) \rangle$*.*

A Gröbner basis of an ideal $\mathcal{I}$ is a useful computational tool for studying the ideal $\mathcal{I}$, in particular dividing a polynomial with a Gröbner basis always gives a unique remainder, allowing one to determine membership in I. We now revise the complexity of computing Gröbner bases.

Definition 18. *A sequence of homogeneous polynomials* $(f_1, \dots, f_m)$ *in* $\mathcal{P}$ *where* $m \leq n$ *is said to be* regular *if for all* $i = 1, \dots, m$ *and* $g \in \mathcal{P}$ *such that* $gf_i \in \langle f_1, \dots, f_{i-1} \rangle$ *we have that* $g \in \langle f_1, \dots, f_{i-1} \rangle$.

Most importantly for us, a sequence $(f_1, \dots, f_m)$ of polynomials over $\mathbb{Q}$ is regular if and only if the Hilbert series of the ideal $\mathcal{I} = (f_1, \dots, f_m)$ is equal to

$$S_{m,n}(z) = \frac{\prod_{i=1}^{m}(1 - z^{d_i})}{(1 - z)^n},$$

where d_i is the total degree of f_i for $i = 1, \dots, m$. The first non-positive coefficient of the above series is called the index of regularity i_{reg} of $\mathcal{I}$. For the rest of this section, we consider polynomials over $\mathbb{Q}$.

The above construction works well for underdetermined and exactly determined systems, but we are mainly interested in the case of overdetermined polynomial systems. Note the ideals we are interested in are zero-dimensional, i.e. the set of roots of the system of equations generating the ideal is finite. The main idea is to only considered regularity up to the index of regularity [3,5,6], which is defined as follows.

Definition 19 ([6]). *Let* $\mathcal{I} = \langle f_1, \dots, f_m \rangle$ *be an ideal of* $\mathcal{P}$. *The* index of regularity *is defined as*

$$i_{reg} = \min \left\{ d \,;\, \dim_{\mathbb{F}}(\{f \in \mathcal{I} \,;\, \deg(f) = d\}) = \binom{n + d - 1}{d} \right\}.$$

The above definition is equivalent to our previous one if $m \leq n$, thus subsuming it. This notion for $m > n$ will be called semi-regularity.

Definition 20 ([6, Def. 4]). *A sequence* $(f_1, \dots, f_m)$ *of homogeneous polynomials in* $\mathcal{P}$ *is* semi-regular *if for all* $i = 1, \dots, m$ *and* $g \in \mathcal{P}$ *such that* $gf_i \in \langle f_1, \dots, f_{i-1} \rangle$ *and* $\deg(gf_i) < i_{reg}$, *we have that* $g \in \langle f_1, \dots, f_{i-1} \rangle$.

For $m \leq n$ the definitions of regularity and semi-regularity coincide. Most importantly for us, a sequence $(f_1, \dots, f_m)$ is semi-regular if and only if the Hilbert series of the ideal $\mathcal{I} = \langle f_1, \dots, f_m \rangle$ is equal to $S_{m,n}(z)$ truncated after the first non-positive coefficient, whose index is precisely i_{reg}. The index of regularity bounds the complexity of Gröbner basis computation in the following way.

Lemma 1 ([6, Prop. 5]). *The total number of field operations in* $\mathbb{F}$ *performed by the matrix version of the* F_5 *algorithm is bounded by*

$$O\left(m i_{reg} \binom{n + i_{reg} - 1}{i_{reg}}^{\omega} \right).$$

As suggested in [6], the same complexity bound holds for non-homogeneous systems if we consider their homogeneous parts of the highest total degree [43, Thms. 1.72 and 1.73]. Moreover, we call a sequence of polynomials semi-regular if the sequence formed by their homogeneous parts of the highest total degree is semi-regular.

3 Lattice Isomorphism as a Group Action

In this section we introduce lattice isomorphism in the quadratic form setting as a group action, and provide some results related to the group action. Consider the equivalence relation $\simeq_\pm$ defined as

$$A \simeq_\pm B \iff A = \pm B,$$

and define the quotient set $\mathrm{GL}_n^\pm(\mathbb{Z}) \coloneqq \mathrm{GL}_n(\mathbb{Z})/\simeq_\pm$. The elements of $\mathrm{GL}_n^\pm(\mathbb{Z})$ are equivalence classes, each containing two elements. Namely, for $A \in \mathrm{GL}_n(\mathbb{Z})$, one has a corresponding class $[A]_\pm \in \mathrm{GL}_n^\pm(\mathbb{Z})$, and $A, -A$ belong to the same class. Define the product between two classes $[A]_\pm, [B]_\pm \in \mathrm{GL}_n^\pm(\mathbb{Z})$ as

$$[A]_\pm \cdot [B]_\pm \coloneqq [BA]_\pm, \tag{1}$$

where BA is the result of the matrix multiplication between two representatives B and A of the classes $[B]_\pm$ and $[A]_\pm$ respectively.

The set $\mathrm{GL}_n^\pm(\mathbb{Z})$ together with the product defined in *Equation* (1) forms a group whose identity element is $[\mathrm{I}_n]_\pm$, the inverse of every element $[A]_\pm \in \mathrm{GL}_n^\pm(\mathbb{Z})$ is $[A^{-1}]_\pm \in \mathrm{GL}_n^\pm(\mathbb{Z})$, and with the associativity property induced by the associativity of matrix multiplication

$$([A]_\pm \cdot [B]_\pm) \cdot [C]_\pm = [BA]_\pm \cdot [C]_\pm = [CBA]_\pm = [A]_\pm \cdot [CB]_\pm = [A]_\pm \cdot ([B]_\pm \cdot [C]_\pm).$$

In what follows, we drop the notation of equivalence classes. Namely, we write $A \in \mathrm{GL}_n^\pm(\mathbb{Z})$ to indicate the class $[A]_\pm \in \mathrm{GL}_n^\pm(\mathbb{Z})$. Within the context of LIP, when we write $U^\mathsf{T} Q U$, we mean the quadratic form obtained by applying any of the two representatives of $[U]_\pm \in \mathrm{GL}_n^\pm(\mathbb{Z})$ (U and $-U$) to $Q \in \mathcal{S}_n^{>0}$. The following proposition defines lattice isomorphisms in the quadratic form representation as a group action over a non-abelian group.

Proposition 1. *Consider a quadratic form $Q \in \mathcal{S}_n^{>0}$ and let $[Q]$ be its equivalence class of isomorphic quadratic forms. Then the map*

$$\star \colon (\mathrm{GL}_n^\pm(\mathbb{Z}) \times [Q]) \to [Q], \qquad \star(V, Q_0) \mapsto V \star Q_0 \coloneqq V^\mathsf{T} Q_0 V,$$

defines a group action of $\mathrm{GL}_n^\pm(\mathbb{Z})$ on $[Q]$.

Proof. Given $Q_0 \in [Q]$ and $V \in \mathrm{GL}_n(\mathbb{Z})$, then $Q_1 = V^\mathsf{T} Q_0 V$ is a quadratic form equivalent to Q_0, and therefore $Q_1 \in [Q]$. The identity element of $\mathrm{GL}_n^\pm(\mathbb{Z})$ fixes, through $\star$, any element of $[Q]$. Finally, for $U, V \in \mathrm{GL}_n^\pm(\mathbb{Z})$ we have that

$$(U \cdot V) \star Q_0 = (VU)^\mathsf{T} Q_0 VU = U^\mathsf{T} (V^\mathsf{T} Q_0 V) U = U \star (V^\mathsf{T} Q_0 V) = U \star (V \star Q_0),$$

which proves compatibility. □

We denote the group action based on lattice isomorphism in the quadratic form representation introduced in Proposition 1 as LIGA. Note that the map $\star$ is defined identically for any class of equivalent quadratic forms $[Q]$. Differently from most other cryptographic group actions used in the literature [1,33,15], in our case we have that both the base set and the group are infinite.[7]

We obtain the following corollary which characterizes the orbits and stabilizers of LIGA.

Corollary 1. *(a) The orbits of* LIGA *are quadratic form equivalence classes.*
(b) The stabilizer of a quadratic form $Q \in \mathcal{S}_n^{>0}$ *w.r.t.* LIGA *is* $\mathrm{Aut}\,(Q)$*.*

The above corollary combined with Remark 3 implies that the automorphism groups of equivalent quadratic forms have the same size, since conjugation is a bijective map. Likewise, the above proposition combined with the Orbit-Stabilizer Theorem show that the set of isomorphisms between Q and Q' with $Q' = U^T Q U = U \star Q$ is $\{UV \colon V \in \mathrm{Aut}\,(Q)\}$, as this set is precisely the coset UG_Q that gets mapped to Q' by f_Q in Remark 3.

We now focus on the properties of LIGA.

Proposition 2. *Let* $Q \in \mathcal{S}_n^{>0}$ *be the quadratic form for a basis of a lattice* $\mathcal{L}$*. Then the group action* $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ *is transitive and faithful.*

Proof. We first prove transitivity by observing that by Corollary 1, the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ admits a single orbit, implying transitivity by Remark 2.

We now prove the group action is faithful by contradiction. Observe that a group action $(G, X, \star)$ is faithful if and only if for every $g \in G \setminus \{e\}$ there exists $x \in X$ such that $x \neq g \star x$, i.e. there is no group element that fixes every set element. Equivalently, the group action is faithful precisely when the subgroup $N = \bigcap_{x \in X} G_x$ of G, known as the kernel of the group action, is trivial. Since the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ acts on a single orbit by Corollary 1, we have

$$N = \bigcap_{Q' \in [Q]} \mathrm{Aut}\,(Q') = \bigcap_{U \in \mathrm{GL}_n^{\pm}(\mathbb{Z})} U \mathrm{Aut}\,(Q)\, U^{-1},$$

where the last equality follows from the fact that if U maps Q to Q', then $\mathrm{Aut}\,(Q') = \{UVU^{-1} \colon V \in \mathrm{Aut}\,(Q)\} = U\mathrm{Aut}\,(Q)\,U^{-1}$. It follows that N is finite as a subset of $\mathrm{Aut}\,(Q)$, and normal as the intersection of the conjugacy class of $\mathrm{Aut}\,(Q)$. Notice that clearly the normal subgroup $N_{\pm} = \{\pm \mathrm{I}_n\}$ of $\mathrm{GL}_n(\mathbb{Z})$ is also contained in N. We thus want to prove that $N/N_{\pm}$ is trivial, i.e. $N = N_{\pm}$.

Now, let us take a non-trivial $U \in N$, i.e. U fixes every element in $[Q]$, let $Q_0 \in [Q]$, and let $Q_1 = V \star Q_0 \neq Q_0$ for $V \neq \pm \mathrm{I}_n \in \mathrm{GL}_n^{\pm}(\mathbb{Z})$. Then, $(UV) \star Q_0 = (VU) \star Q_0$ for every $Q_0 \in [Q]$, which implies $[U, V] = UVU^{-1}V^{-1} \in N$ for all $V \in \mathrm{GL}_n^{\pm}(\mathbb{Z})$. Multiplying by U^{-1} from the left yields that $VUV^{-1} \in N$ for

[7] Notice that LIGA, equipped with $\mathcal{D}_s([Q])$ and an efficient sampler over $\mathrm{GL}_n(\mathbb{Z})$ (Algorithm 1 or **Extract** in [25]), follows the definition of an Effective Group Action from [1], with the relaxation that the base set and group are infinite.

every $V \in \mathrm{GL}_n(\mathbb{Z})$. Therefore, the finite group N contains the entire conjugacy class of U (when viewed in $\mathrm{GL}_n(\mathbb{Z})$), which contradicts the finiteness of N.

Indeed, if the matrix U has a non-zero entry outside of the diagonal, say in the i-th row, then conjugating by V equal to the identity matrix except with $k \in \mathbb{Z}$ in the i-th entry in the first row leads to an infinite subset of the conjugacy class parametrized by $k \in \mathbb{Z}$. In the remaining case of U being a diagonal matrix with ± 1 on the diagonal and not equal to $\pm I$, we see that if U has -1 in the i-th column, then conjugating with V that has a non-trivial i-th row leads to an infinite conjugacy class. □

The following proposition characterizes when the *free* property is satisfied.

Proposition 3. *Let $Q \in \mathcal{S}_n^{>0}$ be a quadratic form. Then, the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is free if and only if Q is automorphism-free.*

Proof. By Corollary 1, orbits are precisely equivalence classes, and because we focus on a single equivalence class $[Q]$, this implies there is a single orbit. By Remark 3, this group action is free if and only if its stabilizers are trivial, which is equivalent by Corollary 1 to the quadratic form Q (or any other orbit representative) being automorphism-free. □

3.1 Cryptographic Properties of LIGA

We introduce in Theorem 1 a new result on the sufficient number of oracle queries to break the weak unpredictability of LIGA. More specifically, we give the necessary number of oracle calls for an adversary to invert the group action in polynomial time and space. Given the generality of the result, we do not limit it to any specific distribution on the group $\mathrm{GL}_n^{\pm}(\mathbb{Z})$ for the secret unimodular matrix. In contrast, we need the distribution on the equivalence class $[Q]$ to satisfy the following property.

Definition 21. *Let $\mathcal{D}_{[Q]}$ be a distribution over $[Q]$, for $Q \in \mathcal{S}_n^{>0}$, and let $d = \frac{n(n+1)}{2}$ and $p \geq d$ be positive integers. We say that $\mathcal{D}_{[Q]}$ induces p-*linear independence *if, given $Q_1, \ldots, Q_p \leftarrow \mathcal{D}_{[Q]}$, the $p \times d$ matrix $M_{\mathcal{Q}}$ whose rows are* $\mathsf{unroll}(Q_i)$ *(see definition in Section 2) is such that*

$$Pr[\mathsf{rank}(M_{\mathcal{Q}}) < d] \leq \mathsf{negl}(n).$$

For simplicity, we write that a distribution $\mathcal{D}_{[Q]}$ is p-linear *when it induces p-linear independence.*

The following proposition follows from Definition 21 since adding a row to a matrix does not decrease its rank.

Proposition 4. *If a distribution $\mathcal{D}_{[Q]}$ over $[Q]$ is p-linear, then it is also $(p+1)$-linear.*

Theorem 1. *Let $Q \in \mathcal{S}_n^{>0}$ and $\mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}$ be a distribution over $\mathrm{GL}_n^{\pm}(\mathbb{Z})$. For $d = \frac{n(n+1)}{2}$, let $\mathcal{D}_{[Q]}$ be a d-linear distribution over $[Q]$. Then the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is not a $(\mathcal{D}_{[Q]}, \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}, t)$-weakly unpredictable group action for any $t \geq d$.*

Proof. We show that $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is not a $(\mathcal{D}_{[Q]}, \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}, d)$-weakly unpredictable group action by providing a polynomial-time algorithm $\mathsf{Recover}$ to invert the group action. Let $\mathcal{A}$ be an adversary able to make $d = \frac{n(n+1)}{2}$ queries to a randomized oracle $F_V^{\$}$ that, when queried, samples a $Q' \leftarrow \mathcal{D}_{[Q]}$ and outputs $(Q', V^{\mathsf{T}} Q' V)$. Then the adversary $\mathcal{A}$ is able to collect a list of d pairs $\mathcal{Q} \coloneqq \{(Q_i, V^{\mathsf{T}} Q_i V)\}_{i=1,\dots,d}$ such that the $d \times d$ matrix $M_{\mathcal{Q}}$ whose rows are composed by $\mathsf{unroll}(Q_i)$ is full-rank with overwhelming probability.

We describe first a procedure $\mathsf{Linearize}$, a sub-routine of the main algorithm $\mathsf{Recover}$ to compute the secret unimodular V. The underlying idea takes inspiration from the work of Rasslan and Youssef [41].

$\mathsf{Linearize}$. Consider one pair $(Q, Q' = V^{\mathsf{T}} Q V)$ from the set $\mathcal{Q}$. Denote with $Q_{i,j}$ (resp. $Q'_{i,j}$) the (i,j)-th entry of Q (resp. Q'). Given that Q is symmetric, we have that $Q_{i,j} = Q_{j,i}$ (resp. $Q'_{i,j} = Q'_{j,i}$). Then, we can write the equation

$$Q'_{i,j} = \sum_{k=1}^{n} \sum_{l=1}^{n} Q_{k,l} \cdot X_{(i,k),(j,l)} \tag{2}$$

where $X_{(i,k),(j,l)} = V_{i,k} \cdot V_{j,l}$ for each $i,j,k,l \in \{1,\dots,n\}$, and $V_{i,j}$ is the (i,j)-th entry of V. Let us consider as baseline Equation (2) with $i = j$:

$$Q'_{i,i} = \sum_{k=1}^{n} \sum_{l=k+1}^{n} 2Q_{k,l} \cdot X_{(i,k),(i,l)} + \sum_{k=1}^{n} Q_{k,k} \cdot X_{(i,k),(i,k)}.$$

Writing the above equation as a d-dimensional vector-matrix multiplication, we get $Q'_{i,i} = \mathbf{Q} \cdot \mathbf{x}_i$ where

$$\mathbf{Q} = \left[Q_{1,1}\ 2Q_{1,2} \dots 2Q_{1,n}\ Q_{2,2}\ 2Q_{2,3} \dots 2Q_{2,n} \dots Q_{n,n}\right], \text{ and}$$
$$\mathbf{x}_i = \left[X_{(i,1),(i,1)} \dots X_{(i,1),(i,n)}\ X_{(i,2),(i,2)} \dots X_{(i,2),(i,n)} \dots X_{(i,n),(i,n)}\right]^{\mathsf{T}}.$$

Denote by $\mathsf{diag}(Q') = \left[Q'_{1,1}\ Q'_{2,2} \dots Q'_{n,n}\right]$ the diagonal of the matrix Q' represented as a vector. Then we have

$$\underset{\substack{\downarrow \\ d\text{-dimensional vector}}}{\mathbf{Q}} \cdot \overbrace{\left[\mathbf{x}_1\ \mathbf{x}_2 \dots \mathbf{x}_n\right]}^{d\text{-by-}n\text{ matrix}} = \mathsf{diag}(Q'). \tag{3}$$

$\mathsf{Recover}$. The procedure $\mathsf{Linearize}$ generates a linear system with d^2 variables and d equations. Given that we have d pairs (Q_i, Q'_i) in $\mathcal{Q}$, we repeat the above technique d times to derive d^2 linearly independent equations, and proceed by finding the unique solution (up to a sign) of the associated system. We describe the algorithm to recover $\pm V$ below and refer to it as $\mathsf{Recover}$:

1. For each pair (Q_i, Q'_i) in $\mathcal{Q}$, apply $\mathsf{Linearize(Q_i, Q'_i)}$ and get the following equation
$$\mathbf{Q}_i \cdot \begin{bmatrix}\mathbf{x}_1 \; \mathbf{x}_2 \; \ldots\ldots \; \mathbf{x}_n\end{bmatrix} = \mathsf{diag}(Q'_i).$$
2. Solve the linear system
$$\begin{bmatrix}\mathbf{Q}_1 \\ \vdots \\ \mathbf{Q}_d\end{bmatrix} \cdot \begin{bmatrix}\mathbf{x}_1 \; \mathbf{x}_2 \; \ldots\ldots \; \mathbf{x}_n\end{bmatrix} = \begin{bmatrix}\mathsf{diag}(Q'_1) \\ \vdots \\ \mathsf{diag}(Q'_d)\end{bmatrix} \tag{4}$$
using Gaussian Elimination to retrieve $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$.
3. Derive the entries (up to a sign) of the solution matrix U by computing first $U_{1,i} = \sqrt{\mathbf{x}_{1,i}}$, then $U_{j,i} = \frac{\mathbf{x}_{j,i}}{U_{1,i}}$ for $0 \leq i, j \leq n$. We have the following two scenarios:
   (a) If $U_{1,i} \neq 0$ for $i = 1, \ldots, n$ then we get (up to a sign) the i-th column V_i of V. That gives 2^n possible combinations, out of which only 2 are correct. From the perspective of the columns of V, each pair (Q, Q') in $\mathcal{Q}$ satisfies
$$V_j Q V_i = Q'_{i,j} \qquad i \leq j, \tag{5}$$
   for each $j \coloneqq 1, \ldots, n$. In other words, Equation (5) describes the inner products $\langle V_j, V_i \rangle_Q$ in the geometry given by Q.
   Thus, to get around this exponential step, we interpret them as column solution parity equations. We guess a solution for U_1, which is the first column of either V or $-V$. Then for each $i \coloneqq 2, \ldots, n$, we pick the solution for U_i such that $U_i Q U_1 = Q'_{1,i}$, since the inner product given by Q is linear in both components. We thus obtain either V or $-V$, depending on our guess of the solution for U_1, after which the algorithm terminates.
   (b) If $U_{1,i} = 0$ for some $1 \leq i \leq n$, then the algorithm cannot recover the full matrix U as it would have to divide by zero. In this case, the algorithm samples a unimodular matrix R using Algorithm 1 for a parameter $T = O(n)$, and computes the set $\mathcal{Q}' \coloneqq \{(Q, R^\mathsf{T} Q' R) \colon (Q, Q') \in \mathcal{Q}\}$, then repeats $\mathsf{Recover}$ with $\mathcal{Q}'$ as input. Note that $M_{\mathcal{Q}'} = M_{\mathcal{Q}}$, and so $\mathsf{rank}(M_{\mathcal{Q}'}) = d$. If the algorithm succeeds in recovering the matrix $U = VR$ (i.e., U has only non-zero entries in its first row), then it also recovers $\pm V$ as $V = UR^{-1}$, after which the algorithm terminates. Otherwise, the algorithm tries again with a different unimodular matrix R until it succeeds.

Memory and time complexities. Recall that $d = \frac{n(n+1)}{2}$. Step 2 from Recover requires to solve the linear system determined by Equation (4), which has a cost of $O(d^\omega) = O(n^{2\omega})$ operations. The derivation of the entries of U in Step 3 takes $O(n^2)$ integer operations. The calculations of the correct sign of the i-th column of U require $2n$ inner product calculations, which gives a total cost of $O(2n^3)$ operations. Then, the time complexity of deriving $\pm V$ becomes

$$O\left(\frac{n^\omega (n+1)^\omega}{2^\omega} + 2n^3 + n^2\right) = O(n^{2\omega})$$

operations. In terms of memory, the algorithm Recover stores one $d \times d$ matrix, two $d \times n$ matrices, and the $n \times n$ matrix U. Therefore, Recover has a memory complexity of storing

$$d^2 + 2dn + n^2 = \frac{n^2 (n+1)^2}{4} + \frac{n^2(n+1)}{2} + n^2 = O(n^4)$$

matrix entries.

We are left to show that the number of tries in Step 3b in Recover is negligible and does not grow with n. Let $R_{1,1}, \ldots, R_{1,n}$ denote the entries of the first row of R which are uniformly distributed in $[-T, T] \subset \mathbb{Z}$ (because of Algorithm 1). We then have that VR has one or more zeros in its first row if and only if $(R_{1,1}, \ldots, R_{1,n})$ is a solution to the Diophantine equation

$$V_{1,j} x_1 + V_{2,j} x_2 + \cdots + V_{n,j} x_n = 0 \tag{6}$$

for some $1 \leq j \leq n$. Since V is non-singular, at least one entry per row is non-zero. Without lost of generality assume $V_{n,j} \neq 0$. Then

$$x_n = -\frac{V_{1,j}}{V_{n,j}} x_1 - \frac{V_{2,j}}{V_{n,j}} x_2 - \cdots - \frac{V_{n-1,j}}{V_{n,j}} x_{n-1},$$

i.e. x_n is uniquely determined by $x_1, \ldots, x_{n-1}$, and whether or not $(x_1, \ldots, x_{n-1})$ leads to a solution or not is determined by a congruence condition modulo $V_{n,j}$. Thus, for every j there exists a rational constant $0 \leq \gamma_j \leq 1$ such that the number of solutions is asymptotic to $\gamma_j (2T+1)^{n-1}$. Therefore, the proportion of solutions on all the possible vectors is asymptotic to $\gamma_j/(2T+1)$. Hence, the probability that $[R_{1,1} \ldots R_{1,n}]$ is not a solution of any of Equation (6) is at least

$$\left(1 - \frac{1}{2T+1}\right)^n = \left(1 - \frac{1}{O(n)}\right)^n = \left(1 - \frac{1}{cn}\right)^n \xrightarrow{n \to \infty} e^{-1/c}$$

for some constant $c \geq 1$. □

Weak pseudorandomness of a permutation is a stronger property than weak unpredictability, therefore we obtain the following corollary.

Corollary 2. *Let $Q \in \mathcal{S}_n^{>0}$ and $\mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}$ be a distribution over $\mathrm{GL}_n^{\pm}(\mathbb{Z})$. For $d = \frac{n(n+1)}{2}$, let $\mathcal{D}_{[Q]}$ be a d-linear distribution over $[Q]$. Then, the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is not a $(\mathcal{D}_{[Q]}, \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}, t)$-weakly pseudorandom group action, for any $t \geq d$.*

Theorem 1 and Corollary 2 also extend to the case of $\mathcal{D}_{[Q]}$ being p-linear for $p > d$, due to Proposition 4.

On d-linear distributions and experimental verification. We believe that the hypothesis on the distribution $\mathcal{D}_{[Q]}$ to be d-linear is realistic. Essentially, we require $\mathcal{D}_{[Q]}$ to output quadratic forms that are linearly independent from each other via the function $\mathsf{unroll}()$. On the other hand, a distribution that outputs samples that are somewhat more likely to be linearly dependent would make them more predictable. Hence, it would likely come with serious security implications when used to build cryptographic primitives.

We were not able to prove that $\mathcal{D}_s\left([Q]\right)$ (described in Definition 5, introduced and used in [25]) is d-linear theoretically. However, we experimentally observed that $\mathcal{D}_s\left([Q]\right)$ behaves as a d-linear distribution. Therefore, we make the following assumption.

Assumption 1 *For a quadratic form $Q \in \mathcal{S}_n^{>0}$, the Gaussian Form Distribution $\mathcal{D}_s([Q])$ with $s \geq \max\{\lambda_n(Q), \left\|B_Q^*\right\| \cdot \sqrt{\ln(2n+4)/\pi}\}$ is $\frac{n(n+1)}{2}$-linear.*

Using $\mathcal{D}_s\left([Q]\right)$ as distribution for the base set $[Q]$, we verified the correctness of $\mathsf{Recover}$ presented in the proof of Theorem 1 via a SageMath implementation available at [9].

On commutative subgroups of $\mathrm{GL}(\mathbb{Z})$**.** If the secret unimodular matrix belongs to a commutative subgroup of $\mathrm{GL}(\mathbb{Z})$ (e.g. circulant matrices, powers of a matrix, ...), then it can be recovered in polynomial time from one single sample. More precisely, let $\mathcal{G}_c \subset \mathrm{GL}(\mathbb{Z})$ be a commutative group, and let $V \in \mathcal{G}_c$. Given a LIP instance $(Q, Q' = V^{\mathsf{T}}QV)$, one is able to construct more LIP instances sharing the same secret unimodular matrix V (and simulate the calls to the oracle in Theorem 1) as follows. Sample unimodular matrices $U \in \mathcal{G}_c$ and compute

$$(\bar{Q} \coloneqq U^{\mathsf{T}}QU,\ \bar{Q}' \coloneqq U^{\mathsf{T}}Q'U = U^{\mathsf{T}}V^{\mathsf{T}}QVU = V^{\mathsf{T}}U^{\mathsf{T}}QUV = V^{\mathsf{T}}\bar{Q}V).$$

Hence, from one single call to the oracle, one can efficiently generate a long enough list of LIP instances sharing the same secret unimodular V and use $\mathsf{Recover}$ described in the proof of Theorem 1 to retrieve it.

4 Time/Samples Trade-off Using Gröbner Basis

In this section we propose another approach for computing the secret unimodular matrix V from a list of m pairs $\mathcal{Q} = \{(Q_i, Q_i' = V^TQ_iV)\}_{i=1,\dots,m}$ given by the randomized oracle $F_V^{\$}$ using Gröbner bases. This approach allows one to use fewer samples, i.e. take $m \leq d$, at the price of an increased computational complexity, and later allows us to target weak pseudorandomness specifically.

4.1 Algebraic Analysis

Naively, one may define n^2 variables $\{x_{i,j}\}_{i,j=1,\dots,n}$ representing the individual entries of V, and think of each sample of the form

$$Q' = \begin{bmatrix} x_{1,1} & \cdots & x_{n,1} \\ \vdots & \ddots & \vdots \\ x_{1,n} & \cdots & x_{n,n} \end{bmatrix} \cdot Q \cdot \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix}$$

as giving $d = \frac{n(n+1)}{2}$ quadratic equations. However, notice the equations given are structured, as each only contains either n or $2n$ variables. Indeed, as noted already in the proof of Theorem 1 what we are given by the oracle $F_V^{\$}$ are inner product equations for the columns of V as in Equation (5). In particular, looking at $i = j$ each sample yields one norm equation $||V_i||_Q^2 = Q'_{i,i}$ per column, containing merely n variables.

Remark 4. Using the same observations in both the linearization approach and the algebraic approach using Gröbner basis computation is no coincidence. Algebraically, linearization is no more than the interreduction of a system of equations, i.e. reducing each polynomial of the system w.r.t. all others, whereas a Gröbner basis algorithm would also compute S-polynomials.

The main idea is the following. Instead of using all the equations given by the oracle $F_V^{\$}$, we focus on collecting norm equations to obtain m quadratic equations in n variables, one system per column of the secret unimodular matrix V. If these systems are sufficiently random, which we qualify in Assumption 2, each system will have two unique solutions $\pm V_i$ with overwhelming probability. We then use the inner product equations (from a single oracle query) to assemble these solutions into $\pm V$, which act equivalently on quadratic forms. Note also that the systems for different columns are disjoint in terms of variables, meaning the solutions for each column can be computed independently.

We require the following of our quadratic systems of equations.

Assumption 2 *For a quadratic form $Q \in \mathcal{S}_n^{>0}$, a unimodular matrix V, and $\{Q_i\}_{i=1,\dots,m}$ with $m > n$ sampled from the Gaussian Form Distribution $\mathcal{D}_s([Q])$ with $s \geq \max\{\lambda_n(Q), \left\|B_Q^*\right\| \cdot \sqrt{\ln(2n+4)/\pi}\}$, the system of norm equations obtained from $\mathcal{Q} = \{(Q_i, V^T Q_i V)\}_{i=1,\dots,m}$ forms a semi-regular sequence for each column of V.*

As the authors of [6] point out in their conclusion, this assumption (when the polynomials are viewed over $\mathbb{Q}$) is another form of Fröberg's conjecture [28], and it seems to hold experimentally.

Since computing each column is independent from all other column, we focus our analysis on the quadratic system of norm equations for one column. We have n as the number of variables, denote by $m > n$ the number of equations, and for each equation the total degree is $d_i = 2$. Under Assumption 2, the Hilbert series of the system is given by the following expression

$$S_{m,n}(z) = \frac{\prod_{i=1}^{m}(1-z^2)}{(1-z)^n} = \frac{(1-z)^m(1+z)^m}{(1-z)^n} = (1-z)^{m-n}(1+z)^m$$

The computational complexity is by Lemma 1 exponential in i_{reg}, but is of course polynomial in m and n for any fixed i_{reg}. We note that the analysis in e.g. [6] is concerned with the behaviour of i_{reg} when the relationship between m and n is determined. We instead treat i_{reg} as fixed by the adversary depending on their computational power, and analyse how many equations are needed. The index of regularity i_{reg} is essentially determined by how many equations we have, descending towards 2 as m approaches n^2, and each sample gives precisely one norm equation for this column. We now analyse how many equations we need to reach the smallest possible $i_{\mathrm{reg}} = 2$, where the Gröbner basis computation amounts only to producing S-polynomials of degree at most 2 and then using linear algebra on the resulting Macaulay matrix [34].

In order to achieve this, we need the coefficient c_2 in front of z^2 to be non-positive in $S_{m,n}(z)$. We therefore want

$$c_2 = \binom{m}{2} - \binom{m}{1}\binom{m-n}{1} + \binom{m-n}{2} \leq 0$$

which simplifies to

$$\frac{n(n+1)}{2} \leq m,$$

meaning we need $m = d$ equations to reach $i_{\mathrm{reg}} = 2$, the same amount of samples we need for the linearization approach. The complexity of the Gröbner basis computation for a single column is then bounded by

$$O\left(2d\binom{n+1}{2}^{\omega}\right) = O\left(d^{1+\omega}\right) = O\left(n^{2+2\omega}\right).$$

By allowing the index of regularity i_{reg} to grow, we can reduce the number of equations (and thus samples) needed. A similar analysis for allowing $i_{\mathrm{reg}} = 3$ requires that

$$c_3 = \binom{m}{3} - \binom{m}{2}\binom{m-n}{1} + \binom{m}{1}\binom{m-n}{2} - \binom{m-n}{3} \leq 0$$

which simplifies to

$$\frac{n^2+3n+2}{6} \leq m,$$

while the complexity of Gröbner basis computation for a single column will be bounded by

$$O\left(3m\binom{n+2}{3}^{\omega}\right) = O\left(n^{2+3\omega}\right).$$

The following statement captures the approximate relation between the number of samples needed and the growth in complexity.

Proposition 5. *For any index of regularity $i \geq 2$, at least $m = O\left(\frac{n^2}{i^2}\right)$ oracle queries of $F_V^{\$}$ are required to compute the secret unimodular matrix V using Gröbner bases, with computational complexity bounded by $O(n^{2+i\omega})$.*

Proof. The number of samples for a fixed index of regularity is determined by

$$c_i = \sum_{k=1}^{i} (-1)^k \binom{m-n}{k} \binom{m}{i-k} = 0,$$

as given by the Binomial Theorem. Notice the highest degree term in n is $\frac{n^i}{i!}$, and that all degree-i terms that contain m cancel. The highest degree term in n that contains m is then $-\ell m n^{i-2}$ for a small positive rational constant ℓ depending only on i.

We rearrange the above equation by keeping all terms that contain only the variable n on the left hand side, and moving all other terms to the right hand side. We get an equation of the form

$$\frac{n^i}{i!} + O\left(n^{i-1}\right) \approx \ell m n^{i-2} \cdot O(1)$$

once we replace all the variables m inside the parentheses on the right hand side with their approximation n^2. Dividing both sides by ℓn^{i-2} we get $\frac{n^2}{\ell i!} \approx m$.

Let us consider the factor ℓ in its relation to i. The only terms of c_i that contribute to the coefficient in front of mn^{n-2} are terms with $k \in \{i, i-1, i-2\}$. The contribution of $k = i$ is

$$(-1)^i (-1)^{i-2} \frac{(-1-2-\cdots-(i-1))(i-1)}{i!} = -\frac{i(i-1)^2}{2i!},$$

the contribution of $k = i-1$ is

$$(-1)^{i-1} (-1)^{i-2} \frac{-1-2-\cdots-(i-2)}{(i-1)!} = \frac{i(i-1)(i-2)}{2i!},$$

and the contribution of $k = i-2$ is $-(-1)^{i-2}\frac{i(i-1)}{2i!}$. Observe the terms with i^3 in the numerator cancel, hence $\ell \approx \frac{i^2}{i!}$, and we get that $m \approx \frac{n^2}{i^2}$.

The complexity bound for a fixed i then follows from the bound given by Lemma 1

$$O\left(im\binom{n+i-1}{i}^{\omega}\right) = O\left(n^{2+i\omega}\right),$$

noting that $\binom{n+i-1}{i}$ is a polynomial of degree i in n. □

Remark 5. We provide a table of approximate numbers of required samples for small values of i_{reg}, which shows the proposition holds even for small i_{reg}, not just asymptotically.

i_{reg}	2	3	4	5	6
$\frac{m}{n^2} \approx$	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{12}$	$\frac{1}{20}$	$\frac{1}{30}$

Table 1: Approximate relationship between the number of queries of $F_V^{\$}$ needed to reach a desired index of regularity i_{reg} and n^2.

Finally, we describe our algorithm in the following theorem.

Theorem 2. *Let $Q \in \mathcal{S}_n^{>0}$, $i_0 \in \mathbb{N}$, and $\mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}$ be a distribution over $\mathrm{GL}_n^{\pm}(\mathbb{Z})$. Let $m \leq d$ be the number of oracle queries of $F_V^{\$}$ such that the systems of column norm equations have $i_{reg} \leq i_0$. Then the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is not $(\mathcal{D}_s([Q]), \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}, t)$-weakly unpredictable for any $t \geq m$.*

Proof. We again show that $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is not a $(\mathcal{D}_{[Q]}, \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}, d)$-weakly unpredictable group action by providing a polynomial-time algorithm RecoverGB to invert the group action. Let $\mathcal{A}$ be an adversary able to make m queries to a randomized oracle $F_V^{\$}$ that, when queried, samples a $Q' \leftarrow \mathcal{D}_s([Q])$ and outputs $(Q', V^{\mathsf{T}}QV)$. The strategy of the adversary is the following.

By querying the oracle $F_V\$$, the adversary $\mathcal{A}$ is able to collect a list of m pairs $\mathcal{Q} \coloneqq \{(Q_i, V^{\mathsf{T}}Q_iV)\}_{i=1,\dots,m}$, and extracts the column norm equations as follows. The adversary keeps n lists, one for each column. When receiving a pair (Q, Q') from the oracle $F_V^{\$}$, they save the quadratic equation

$$\begin{bmatrix} x_1 \cdots x_n \end{bmatrix} \cdot Q \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} - Q'_{i,i} = 0$$

to their i-th list. They only need to keep the inner product equations of the last sample to extract the complete solution.

The adversary $\mathcal{A}$ then runs RecoverGB that computes the reduced Gröbner basis w.r.t. the DRL monomial order of the ideal generated by each system of equations and obtains two solutions for each, which are exactly the i-th columns of V and $-V$, which we denote by $\pm V_i$.

To extract the matrix V, the algorithm RecoverGB then proceeds by guessing the solution $\tilde{V}_1$ for the first column. For all $i = 2, \dots, n$ it then picks $\tilde{V}_i \in \{\pm V_i\}$ such that $\tilde{V}_1 Q \tilde{V}_i = Q'_{1,i}$, where (Q, Q') is the last sample that $\mathcal{A}$ kept. Depending on the guess $\tilde{V}_1$, the algorithm has computed either V or $-V$, allowing the adversary $\mathcal{A}$ to reproduce the group action of V. □

4.2 Weak Pseudorandomness

When approaching weak pseudorandomness of LIGA alone, we can optimize the approach further both for the number of samples needed and the complexity estimate. Since the adversary is now not required to find the secret unimodular matrix V (or at least be able to reproduce its action on quadratic forms), it is enough for them to show that merely one system of norm equations describing one column has a solution. Furthermore, since breaking weak pseudorandomness is a decision problem, the adversary can consider solving the system of equations not over $\mathbb{Q}$ (or $\mathbb{Z}$) but a small finite field, namely $\mathbb{F}_2$ where they can use field equations $x_i^2 - x_i = 0$ of degree 2 for all $i = 1, \ldots, n$. Observe that the solution over $\mathbb{F}_2$ is unique, implying also there is no need to extract the final solution from column solutions.

Remark 6. An interesting consequence of this approach is how it positions LIGA as a cryptographic group action in relation to other hard computational problems, namely the $\mathcal{MQ}$ problem. Indeed, if the Gaussian Form Distribution $\mathcal{D}_s([Q])$ is wide enough [37, Lemmas 3.2 and 3.3], i.e. exceeds the Smoothing Bound, then by the Smoothing Lemma [37,29], the coefficients of the system of equations modulo 2 may be treated as uniformly random, since the two distributions are statistically close. It follows that if an adversary is able to solve $\mathcal{MQ}$ over $\mathbb{F}_2$ with n variables and m equations efficiently then they can also efficiently break weak pseudorandomness of the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ using m samples for any $Q \in \mathcal{S}_n^{>0}$. More specifically, in this case the adversary is solving the Boolean $\mathcal{MQ}$ problem studied in detail in [7].

The approach is now twofold. We can compute over $\mathbb{Q}$ like in Section 4.1 and include the field equations into each of our column norm systems of equations which we reduce modulo 2. Since there are n field equations and all have degree 2, our analysis from above stands and we require $d - n = \frac{n(n-1)}{2}$ oracle queries of $F_V^{\$}$ to reach $i_{\mathrm{reg}} = 2$, as if we only had $n-1$ variables. More generally, if reaching the index of regularity $i_{\mathrm{reg}} = i$ over $\mathbb{Q}$ requires m oracle queries, then solving the same system modulo 2 over in $\mathbb{Q}$ with the $\mathbb{F}_2$ field equations added requires $m - n$ oracle queries.

Alternatively, one may exclude the field equations from the m equations and instead compute over $\mathbb{F}_2$ directly. The field equations still need to be implied so as not to end up in the algebraic closure of $\mathbb{F}_2$, i.e. we are computing in the quotient ring $\mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle$. Following the analysis in [4, Cor. 7] and assuming the systems of equations remain semi-regular sequences in $\mathbb{F}_2$ as implied in Remark 6, the Hilbert series of our system of m equations and n variables is

$$T_{m,n}(z) = \frac{(1+z)^n}{(1+z^2)^m}$$

truncated after the first non-positive coefficient, which expands (around 0) to

$$T_{m,n}(z) = 1 + nz + \left(\frac{(n-1)n}{2} - m\right) z^2 + \left(\frac{(n-2)(n-1)n}{6} - mn\right) z^3 + O(z^4),$$

implying identical bounds for $i_{\mathrm{reg}} = 2, 3$ as in our analysis above, i.e. we need n fewer equations, courtesy of field equations. The Gröbner basis computation complexity estimate of $O(mi_{\text{reg}}\binom{n}{i_{\text{reg}}})$ given by [6, Prop. 9] then implies the bound of $O(n^{2+2\omega})$ (for one column) when $i_{\text{reg}} = 2$, and $O(n^{2+i_{\text{reg}}\omega})$ more generally.

Corollary 3. *Let $Q \in \mathcal{S}_n^{>0}$, $i_0 \in \mathbb{N}$, and $\mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}$ be a distribution over $\mathrm{GL}_n^{\pm}(\mathbb{Z})$, and let $s \geq \max\{\lambda_n(Q), ||B_Q^*|| \cdot \sqrt{\ln(2n+4)/\pi}\}$. Let $m \leq d - n$ be the number of oracle queries of $F_V^{\$}$ such that the systems of column norm equations modulo 2 have $i_{reg} \leq i_0$. Then* LIGA *is not a $(\mathcal{D}_s([Q]), \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z})}, t)$-weakly pseudorandom group action for any $t \geq m$.*

Proof. Following the reasoning in Remark 6, the parameter s is large enough by the Smoothing Bound [25, Lemma 2.6] so that the distribution ($\mathcal{D}_s([Q])$ mod 2), which returns the unimodular matrix U and the quadratic form $Q' = U^T QU$ both with entries modulo 2, is within negligible statistical distance of the uniform distribution on respective sets modulo 2. The strategy of the adversary is therefore identical to the one presented in the proof of Theorem 2, except that they store equations modulo 2, and compute the Gröbner bases of the ideals generated by the column norm systems of equations in the quotient ring $\mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle$ instead of over $\mathbb{Q}$. □

4.3 Comparisons and Experimental Results

This section conducts a comparative analysis between the linearization (see proof of Theorem 1) and the Gröbner basis (see proof of Theorem 2) approaches to illustrate the trade-off between running time and required number of samples to recover the secret. We provide a proof-of-concept implementation (using the Sage Mathematics Software System SageMath [45]) available at [9]. Our implementation of the Gröbner basis approach uses the msolve library for solving polynomial systems [10].

By applying the Gröbner basis technique we can significantly reduce the number of samples required to break weak unpredictability, from $m = \frac{n(n+1)}{2}$ to $m \approx \frac{n^2}{i^2}$ for any constant index of regularity $i \geq 2$, while still ensuring a polynomial running time, whereas recovering the secret using linearization with fewer than $d = \frac{n(n+1)}{2}$ samples is not possible. However, the required time complexity of computing Gröbner bases to recover the secret with fewer samples is notably higher than that of the linearization approach in both a practical and asymptotic sense. Table 2 presents an asymptotic comparison between Gröbner basis and linearization techniques regarding the number of samples and runtime.

One could try to reduce the number of equations needed to recover the secret even lower at the cost of even higher complexity. For example, for $m = n\log_2(n)$ the complexity of Gröbner basis computation is known to be sub-exponential [4], however for smaller examples (e.g. $n = 32$) it can be done with index of regularity 3. This is because $\frac{n^2}{i^2}$ and $n\log_2(n)$ meet for small n with increasing i. Once n becomes larger, the index of regularity quickly grows as well (e.g. for $n = 64$ we require $i_{\text{reg}} = 4$). Therefore, for cryptographically relevant sizes, the Gröbner basis approach is able to reduce the number of samples only by a small constant

n	16	32	64	128	256	512	1024
Recover	22.5	28.1	33.7	39.3	44.9	50.5	56.2
RecoverGB with $i_{\text{reg}} = 2$	30.5	38.1	45.7	53.3	60.9	68.5	76.2
RecoverGB with $i_{\text{reg}} = 3$	41.7	52.1	62.5	73.0	83.4	93.8	104.2
RecoverGB with $i_{\text{reg}} = 4$	52.9	66.2	79.4	92.6	105.8	119.1	132.3
RecoverGB with $i_{\text{reg}} = 5$	64.2	80.2	96.2	112.3	128.3	144.3	160.4

Table 2: Asymptotic comparison between Gröbner basis and linearization regarding runtime. We consider Strassen's algorithm for the complexity exponent $\omega = \log_2(7)$ and present the base two logarithm of the running times. Note that, as the index of regularity (i_{reg}) increases, fewer samples are required to successfully recover the secret unimodular matrix.

as shown in Proposition 5 (since we ideally want the index of regularity to be 2 or 3 as can be seen from Table 2).

We additionally compare the linearization and Gröbner basis approaches through timed experiments, as outlined in Table 3 and Figure 1. All our experiments were conducted on a 2.3 GHz 8-Core Intel Core i9 machine with 16GB of RAM. We emphasize that our implementation must be viewed as a proof-of-concept implementation and, therefore, is not optimized.

n	16	20	24	28	32	36	40
Sampling	13.63	34.68	84.84	504.83	1198.48	2321.70	4652.40
Recover	0.34	1.00	1.98	3.36	5.51	10.57	17.31
RecoverGB	2.04	5.64	13.40	31.59	67.72	130.52	252.16

Table 3: The timings correspond with the average time (in seconds) of eight random LIP instances. In all the experiments, RecoverGB uses $m = n(n+1)/2$ (which has $i_{\text{reg}} = 2$), and no parallelization. The row labeled Sampling corresponds with the timings of generating all the $m = n(n+1)/2$ random instances of LIP with the same fixed secret U.

5 New Hard Problems on Quadratic Forms

In this section, we introduce the following two new hard problems on quadratic forms. We make use of Theorem 1 to provide polynomial-time reductions from sLIP_Q to both problems, for any automorphism-free Q.

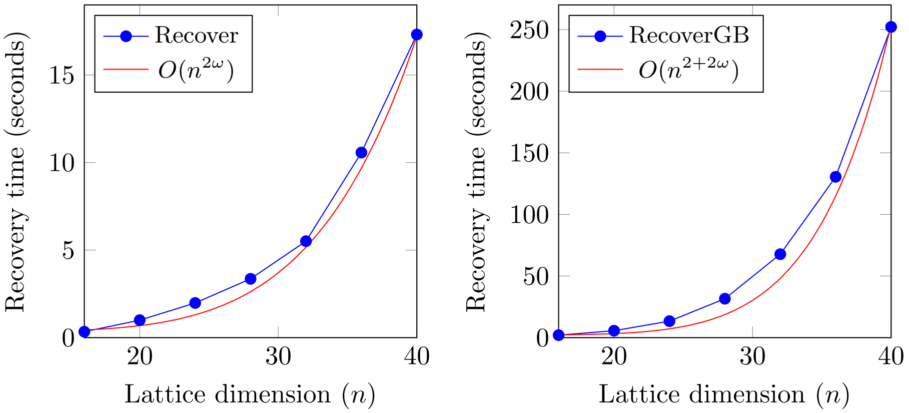

Fig. 1: In the above plots, we interpolate the first and the last measurements for Recover and RecoverGB from Table 3 with the corresponding asymptotic estimations $O(n^{2\omega})$ and $O(n^{2+2\omega})$ for $\omega = \log_2(7)$, and plot all table values. One can see that the experimental measurements fit the theoretical estimations well.

Definition 22 (Transpose Quadratic Form Problem (TQFP)). *Let $\mathcal{L}(B)$ be a full-rank n-dimensional lattice and $Q \in \mathcal{S}_n^{>0}$ be the quadratic form $Q = B^\mathsf{T}B$. Given $Q' \in [Q]$, the Transpose Quadratic Form Problem is to compute $\widehat{Q} \in [Q]$ such that $\widehat{Q} = UQU^\mathsf{T}$, where $U \in \mathrm{GL}_n(\mathbb{Z})$ satisfies $Q' = U^\mathsf{T}QU$.*

Definition 23 (Inverse Quadratic Form Problem (IQFP)). *Let $\mathcal{L}(B)$ be a full-rank n-dimensional lattice and $Q \in \mathcal{S}_n^{>0}$ be the quadratic form $Q = B^\mathsf{T}B$. Given $Q' \in [Q]$, the Inverse Quadratic Form Problem is to compute $\widehat{Q} \in [Q]$ such that $\widehat{Q} = U^{-\mathsf{T}}QU^{-1}$, where $U \in \mathrm{GL}_n(\mathbb{Z})$ satisfies $Q' = U^\mathsf{T}QU$.*

TQFP and IQFP accept as many solutions as the number of isomorphisms between Q and Q', up to the sign. For example, for the case of TQFP, the solution set is defined as $S_{Q'} := \{\widetilde{Q}_V = (VU)^\mathsf{T}Q(VU) \colon V \in \mathrm{Aut}\,(Q)\}$. For the specific case of automorphism-free quadratic forms, the solution is unique ($|S_{Q'}| = 1$). Under Assumption 1, we give in Lemma 2 and Lemma 3 polynomial-time reductions from sLIP_Q to TQFP and IQFP, respectively. We implemented and successfully tested these reductions in a SageMath implementation available at [9].

Lemma 2 (From sLIP_Q to TQFP). *Let $Q \in \mathcal{S}_n^{>0}$ be an automorphism-free quadratic form. Given an oracle $\mathcal{O}_{\mathsf{TQFP}}$ that solves TQFP in time T_0, there is an algorithm that solves sLIP_Q in expected time $O\left(n^2(T_0 + T_1) + n^{2\omega}\right)$, where T_1 is the time complexity of one call to $\mathcal{D}_s\left([Q]\right)$, for $s \geq \max\{\lambda_n(Q), \left\|B_Q^*\right\| \cdot \sqrt{\ln(2n+4)/\pi}\}$.*

Proof. Fixing the same setup as Definition 22, we have Q and $Q' = V^\mathsf{T}QV$, where $Q' \in [Q]$. For simplicity, we assume that $\mathcal{O}_{\mathsf{TQFP}}$ always solves TQFP for isomorphic input Q, Q'. We give an algorithm which solves sLIP_Q with a polynomial number of calls to $\mathcal{O}_{\mathsf{TQFP}}$ as follows. Let us set $d = \frac{n(n+1)}{2}$.

1. Forward (Q', Q) to $\mathcal{O}_{\mathsf{TQFP}}$ and receive the response $\widehat{Q} = VQV^{\mathsf{T}}$.
2. (a) Sample a quadratic form $\bar{Q} = W^{\mathsf{T}}QW$ along with $W \in \mathrm{GL}_n(\mathbb{Z})$ from $\mathcal{D}_s([Q])$.
   (b) Compute $Q'' = W\widehat{Q}W^{\mathsf{T}} = WVQV^{\mathsf{T}}W^{\mathsf{T}}$ and send (Q'', Q) to $\mathcal{O}_{\mathsf{TQFP}}$. Record its response as $\bar{Q} = V^{\mathsf{T}}W^{\mathsf{T}}QWV = V^{\mathsf{T}}\bar{Q}V$.
3. Repeat Step 2 a necessary number of times, for different unimodular W, to derive a set $\mathcal{Q} = \left\{ \left(Q_0^{(i)}, Q_1^{(i)} \right),\ i = 1, \dots, d \right\}$ such that the $d \times d$ matrix $M_{\mathcal{Q}}$ whose rows are $\mathsf{unroll}\left(Q_0^{(i)} \right)$ is full rank.
4. Retrieve $V \leftarrow \mathsf{Recover}(\mathcal{Q})$ as described in Theorem 1.

Running time. Let us assume both matrix multiplication and inversion take $O(n^\omega)$ integer operations. Step 1 costs one call to the oracle $\mathcal{O}_{\mathsf{TQFP}}$. Step 2 samples one random unimodular matrix, makes four matrix multiplications, and two queries to $\mathcal{O}_{\mathsf{TQFP}}$. Now, Step 2 must be repeated $O\left(\frac{n(n+1)}{2}\right)$ times to derive enough linear equations (Step 3). Then Steps 1 to 3 have complexity

$$O\left(T_0 + \frac{n(n+1)}{2} (2T_0 + T_1 + 4n^\omega) \right) = O\left(n^2(T_0 + T_1) + n^{2+\omega} \right).$$

Step 4 requires $O\left(n^{2\omega}\right)$ operations to retrieve V, which gives a total asymptotic time complexity of $O\left(n^2(T_0 + T_1) + n^{2\omega}\right)$.

□

Remark 7. Regarding Lemma 2, in practice one can reduce the number of calls to $\mathcal{O}_{\mathsf{TQFP}}$ by a factor of n by exploiting the following. Let $Q, Q', \widehat{Q} \in \mathcal{S}_n^{>0}$ be equivalent quadratic forms with $Q' = V^{\mathsf{T}}QV$ and $\widehat{Q} = VQV^{\mathsf{T}}$, for some unimodular matrix $V \in \mathrm{GL}_n(\mathbb{Z})$. Then, one can compute the quadratic forms

$$Q_1 \coloneqq Q'QQ' = V^{\mathsf{T}}QVQV^{\mathsf{T}}QV, \qquad Q_0 \coloneqq Q\widehat{Q}Q,$$

and have that (Q_0, Q_1) is such that $Q_1 = V^{\mathsf{T}}Q_0V$. Iteratively, one can define

$$Q_1^{(i)} \coloneqq Q'(QQ')^i, \qquad Q_0^{(i)} \coloneqq Q(\widehat{Q}Q)^i,$$

with $Q_1^{(i)} = V^{\mathsf{T}}Q_0^{(i)}V$, for $i \geq 0$. The Cayley-Hamilton theorem ensures that, for any square matrix M with n rows over a commutative ring, we have that $M^n \in \mathrm{Span}\{I_n, M, M^2, \dots, M^{n-1}\}$ [2, §7.11]. Therefore, with the above approach, we can get a set $\mathcal{Q} = \left\{(Q_i, Q'_i = V^{\mathsf{T}}Q_iV)\right\}_{i=1}^p$ of size $p \leq n$ knowing $Q' = V^{\mathsf{T}}QV$ and $\widehat{Q} = VQV^{\mathsf{T}}$. Using this trick in Step 2 of the proof of Lemma 2, and assuming that p reaches n with high probability, one can reduce the number of calls to $\mathcal{O}_{\mathsf{TQFP}}$ by a factor of n. In this case, taking also into consideration the number of matrix multiplications, the total cost of the reduction in Lemma 2 would be $\widetilde{O}(n(T_0 + T_1) + n^{2\omega})$.[8] In our SageMath implementation, we im-

[8] We have $\widetilde{O}(\cdot)$ instead of $O(\cdot)$ because of the increase of the integer coefficients size when applying this optimization trick.

plemented and tested the variant of the reduction in Lemma 2 that uses such optimization in Step 2.

Lemma 3 (From sLIP_Q to IQFP). *Let $Q \in \mathcal{S}_n^{>0}$ be an automorphism-free quadratic form. Given an oracle $\mathcal{O}_{\mathsf{IQFP}}$ that solves IQFP in time T_0, there exists an algorithm that solves sLIP_Q in expected time $O\left(n^2(T_0+T_1)+n^{2\omega}\right)$, where T_1 is the time complexity of one call to $\mathcal{D}_s\left([Q]\right)$, for $s \geq \max\{\lambda_n(Q), \left\|B_Q^*\right\| \cdot \sqrt{\ln(2n+4)/\pi}\}$.*

Proof. Fixing the same setup as Definition 23, we have Q and $Q' = V^{\mathsf{T}}QV$, where $Q' \in [Q]$. For simplicity, we assume that $\mathcal{O}_{\mathsf{IQFP}}$ always solves IQFP, for a isomorphic input Q, Q'. We give an algorithm which solves sLIP_Q with a polynomial number of calls to $\mathcal{O}_{\mathsf{IQFP}}$ as follows. Let us set $d = \frac{n(n+1)}{2}$.

1. Forward (Q', Q) to $\mathcal{O}_{\mathsf{IQFP}}$ and receive the response $\widehat{Q} = V^{-\mathsf{T}}QV^{-1}$.
2. (a) Sample a quadratic form $\bar{Q} = W^{\mathsf{T}}QW$ along with $W \in \mathrm{GL}_n(\mathbb{Z})$ from $\mathcal{D}_s\left([Q]\right)$.
   (b) Calculate $Z = W^{-1}$.
   (c) Compute $Q'' = Z^{\mathsf{T}}\widehat{Q}Z = Z^{\mathsf{T}}VQV^{\mathsf{T}}Z$ and send (Q'', Q) to $\mathcal{O}_{\mathsf{IQFP}}$. Record its response as $\widetilde{Q} = V^{\mathsf{T}}W^{\mathsf{T}}QWV = V^{\mathsf{T}}\bar{Q}V$.
3. Repeat Step 2 a necessary number of times, for different unimodular W, to derive a set $\mathcal{Q} = \left\{\left(Q_0^{(i)}, Q_1^{(i)}\right),\ i = 1, \ldots, d\right\}$ such that the $d \times d$ matrix $M_{\mathcal{Q}}$ whose rows are $\mathsf{unroll}\left(Q_0^{(i)}\right)$ is full rank.
4. Retrieve $V \leftarrow \mathsf{Recover}(\mathcal{Q})$ as described in Theorem 1.

Running time. The cost analysis is analogous to Lemma 2, with the addition of a matrix inversion in Step 2. However, this is negligible on the total cost of the reduction, that is $O\left(n^2(T_0+T_1)+n^{2\omega}\right)$.

□

To illustrate the above reductions from Lemma 2 and Lemma 3, we simulate the algorithms concerning TQFP and IQFP using a SageMath library; our code is available at [9].

Remark 8. Lemma 2 and Lemma 3 can be generalized to the case of quadratic forms with a non-trivial automorphism group. However, in this case, the solutions to TQFP and IQFP are not unique, but there are as many solutions as the number of automorphisms divided by 2. Consider the case of a TQFP oracle $\mathcal{O}_{\mathsf{TQFP}}$ that returns one of the possible solutions uniformly at random. Then, the algorithm in Lemma 2 would allow retrieving the correct solution only when, for every query to the algorithm, it returns exactly the solution that we are looking for. Therefore, given that we require n correct solutions from $\mathcal{O}_{\mathsf{TQFP}}$, one must repeat on average the whole algorithm $\left(|\mathrm{Aut}\left(Q\right)|/2\right)^n$ times.

6 Open Problems

We believe that studying the Lattice Isomorphism Problem, as well as other computational problems related to isomorphisms or equivalence classes, in the group action framework is an important research direction which will help build a unified theoretical foundation for constructing cryptographic schemes. Below we list some research directions/open problems that arise from our work.

It would indeed be interesting to know the exact number of LIP samples sharing the same secret V that can be given to an adversary while still maintaining the weakly unpredictable nature of the group action. For example, one could to take into account the equation $\det(V) = \pm 1$. However, this should not be very useful since, for an indeterminate V, it is a polynomial in n^2 variables of total degree n (or $2n$, if one decides to square the equation to get rid of the sign ambiguity), and Gröbner basis computation is exponential in the degree of the system. Theorem 2 seems to imply a limit of $m = O(n)$ LIP samples at which solving a "random" system of quadratic equations is known to have complexity exponential in n [6], making it infeasible for the adversary to solve the system of equations using Gröbner basis computation.

Perhaps more relevant to cryptographers, one can ask: Which cryptographic constructions can be securely realized assuming one-wayness and the aforementioned limited version of weak unpredictability (and weak pseudorandomness) to $O(n)$ oracle calls to $F_V^{\$}$?

Another possible direction is to investigate whether an analogous result can be obtained also for group actions stemming from other equivalence problems, e.g. code equivalence or 3-tensor isomorphism. Recently, the authors of [22] studied the code equivalence problem using representation theory and linearization techniques. It will be interesting to study the impact of employing other algebraic methods such as Gröbner bases in this setting. More generally, it would be interesting to investigate whether other group actions also come with a similar limitation on the number of oracle queries allowed to an adversary, and characterize it in concrete terms such as a number of samples.

Acknowledgments. The authors thank Keita Xagawa, Victor Mateu, and Martin R. Albrecht for fruitful discussions on the topic. We also thank Elena Kirshanova and anonymous reviewers for the useful comments on an earlier version of this manuscript. Benjamin Benčina was supported by the EPSRC and the UK Government as part of the grant EP/S021817/1.

References

1. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 411–439. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64834-3_14
2. Apostol, T.M.: Calculus, Vol. II, Multi-Variable Calculus and Linear Algebra. Blaisdell, Waltham, MA (1969)

3. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. Ph.D. thesis, Université Pierre et Marie Curie - Paris VI (2004), https://theses.hal.science/tel-00449609
4. Bardet, M., Faugère, J.C., Salvy, B.: Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. Research Report RR-5049, INRIA (2003), available at https://inria.hal.science/inria-00071534
5. Bardet, M., Faugère, J.C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proc. of International Conference on Polynomial System Solving Paris, November 2004 in honor of Daniel Lazard. pp. 71–75. ICPSS (2004)
6. Bardet, M., Faugère, J.C., Salvy, B., Yang, B.: Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In: 8th International Symposium on Effective Methods in Algebraic Geometry. pp. 1–17. MEGA (2005)
7. Bardet, M., Faugère, J.C., Salvy, B., Spaenlehauer, P.J.: On the complexity of solving quadratic Boolean systems. Journal of Complexity **29**(1), 53–75 (2013). https://doi.org/10.1016/j.jco.2012.07.001
8. Bennett, H., Ganju, A., Peetathawatchai, P., Stephens-Davidowitz, N.: Just how hard are rotations of $\mathbb{Z}^n$? algorithms and cryptography with the simplest lattice. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 252–281. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_9
9. Benčina, B., Budroni, A., Chi-Domínguez, J.J., Kulkarni, M.: `lip-properties`, available at https://github.com/JJChiDguez/lip-properties.git
10. Berthomieu, J., Eder, C., Safey El Din, M.: msolve: A Library for Solving Polynomial Systems. In: Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation. pp. 51–58. ISSAC '21, Association for Computing Machinery, Saint Petersburg, Russia (2021). https://doi.org/10.1145/3452143.3465545, version 0.6.3
11. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient isogeny based signatures through class group computations. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 227–247. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34578-5_9
12. Biasse, J.F., Micheli, G., Persichetti, E., Santini, P.: LESS is more: Code-based signatures without syndromes. In: Nitaj, A., Youssef, A.M. (eds.) AFRICACRYPT 20. LNCS, vol. 12174, pp. 45–65. Springer, Heidelberg (Jul 2020). https://doi.org/10.1007/978-3-030-51938-4_3
13. Blanks, T.L., Miller, S.D.: Generating cryptographically-strong random lattice bases and recognizing rotations of $\mathbb{Z}^n$. In: Cheon, J.H., Tillich, J.P. (eds.) Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021. pp. 319–338. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-030-81293-5_17
14. Bläser, M., Chen, Z., Duong, D.H., Joux, A., Nguyen, N.T., Plantard, T., Qiao, Y., Susilo, W., Tang, G.: On digital signatures based on isomorphism problems: Qrom security, ring signatures, and applications. Cryptology ePrint Archive, Paper 2022/1184 (2022), https://eprint.iacr.org/2022/1184
15. Borin, G., Persichetti, E., Santini, P.: Zero-knowledge proofs from the action subgraph. Cryptology ePrint Archive, Paper 2023/718 (2023), https://eprint.iacr.org/2023/718
16. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 575–584. ACM Press (Jun 2013). https://doi.org/10.1145/2488608.2488680

17. Brassard, G., Yung, M.: One-way group actions. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO'90. LNCS, vol. 537, pp. 94–107. Springer, Heidelberg (Aug 1991). https://doi.org/10.1007/3-540-38424-3_7
18. Budroni, A., Chi-Domínguez, J.J., Kulkarni, M.: Lattice isomorphism as a group action and hard problems on quadratic forms. Cryptology ePrint Archive, Paper 2023/1093 - version 20230724:055703 (2023), https://eprint.iacr.org/archive/2023/1093/1690178223.pdf
19. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 395–427. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03332-3_15
20. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006), https://eprint.iacr.org/2006/291
21. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Undergraduate Texts in Mathematics, Springer International Publishing, 4. edn. (2015). https://doi.org/10.1007/978-3-319-16721-3
22. D'Alconzo, G., Scala, A.J.D.: Representations of group actions and their applications in cryptography. Cryptology ePrint Archive, Paper 2023/1247 (2023), https://eprint.iacr.org/2023/1247
23. De Feo, L., Fouotsa, T.B., Kutas, P., Leroux, A., Merz, S.P., Panny, L., Wesolowski, B.: SCALLOP: Scaling the CSI-FiSh. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 345–375. Springer, Heidelberg (May 2023). https://doi.org/10.1007/978-3-031-31368-4_13
24. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.P.J.: Hawk: Module LIP makes lattice signatures fast, compact and simple. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 65–94. Springer, Heidelberg (Dec 2022). https://doi.org/10.1007/978-3-031-22972-5_3
25. Ducas, L., van Woerden, W.P.J.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Dunkelman and Dziembowski [26], pp. 643–673. https://doi.org/10.1007/978-3-031-07082-2_23
26. Dunkelman, O., Dziembowski, S. (eds.): EUROCRYPT 2022, Part III, LNCS, vol. 13277. Springer, Heidelberg (May / Jun 2022)
27. Felderhoff, J.: Hard Homogenous Spaces and Commutative Supersingular Isogeny based Diffie–Hellman (2019), https://api.semanticscholar.org/CorpusID:252082464
28. Fröberg, R.: An inequality for hilbert series of graded algebras. Mathematica Scandinavica **56**(2), 117–144 (1985)
29. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008). https://doi.org/10.1145/1374376.1374407
30. Gentry, C., Szydlo, M.: Cryptanalysis of the revised NTRU signature scheme. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 299–320. Springer, Heidelberg (Apr / May 2002). https://doi.org/10.1007/3-540-46035-7_20
31. Haviv, I., Regev, O.: On the lattice isomorphism problem. In: Chekuri, C. (ed.) 25th SODA. pp. 391–404. ACM-SIAM (Jan 2014). https://doi.org/10.1137/1.9781611973402.29
32. Ji, Z., Qiao, Y., Song, F., Yun, A.: General linear group action on tensors: A candidate for post-quantum cryptography. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part I. LNCS, vol. 11891, pp. 251–281. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-36030-6_11

33. Joux, A.: Mpc in the head for isomorphisms and group actions. Cryptology ePrint Archive, Paper 2023/664 (2023), https://eprint.iacr.org/2023/664
34. Lazard, D.: Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In: van Hulzen, J.A. (ed.) Computer Algebra. Lecture Notes in Computer Science, vol. 162, pp. 146–156. Springer Berlin Heidelberg (1983). https://doi.org/10.1007/3-540-12868-9_99
35. Machì, A.: Groups. Springer Milano (2012). https://doi.org/10.1007/978-88-470-2421-2
36. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective, vol. 671. Springer Science+Business Media, LLC (01 2002). https://doi.org/10.1007/978-1-4615-0897-7
37. Micciancio, D., Regev, O.: Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM Journal on Computing **37**(1), 267–302 (2007). https://doi.org/10.1137/s0097539705447360
38. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th FOCS. pp. 458–467. IEEE Computer Society Press (Oct 1997). https://doi.org/10.1109/SFCS.1997.646134
39. NIST: Post-quantum cryptography: Digital signature schemes. https://csrc.nist.gov/projects/pqc-dig-sig
40. NIST: Post-quantum cryptography standardization. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022
41. Rasslan, M.M.N., Youssef, A.M.: Cryptanalysis of a Public Key Encryption Scheme Using Ergodic Matrices. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **94-A**(2), 853–854 (2011). https://doi.org/10.1587/transfun.E94.A.853
42. Serre, J.P.: A Course in Arithmetic, Graduate Texts in Mathematics, vol. 7. Springer New York (1973). https://doi.org/10.1007/978-1-4684-9884-4
43. Spaenlehauer, J.P.: Résolution de Systèmes Multi-homogènes et Déterminantiels. Ph.D. thesis, Université Pierre et Marie Curie (2012)
44. Tang, G., Duong, D.H., Joux, A., Plantard, T., Qiao, Y., Susilo, W.: Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In: Dunkelman and Dziembowski [26], pp. 582–612. https://doi.org/10.1007/978-3-031-07082-2_21
45. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.1) (2023), `https://www.sagemath.org`