

Lattice Isomorphism as a Group Action and Hard Problems on Quadratic Forms

Alessandro Budroni, Jesús-Javier Chi-Domínguez, Mukul Kulkarni

Cryptography Research Center, Technology Innovation Institute, Abu Dhabi, UAE
{alessandro.budroni,jesus.dominguez,mukul.kulkarni}@tii.ae

Abstract. Group actions have been used as a foundation in Public-key Cryptography to provide a framework for hard problems and assumptions. In this work we formalize the Lattice Isomorphism Problem (LIP) within the context of cryptographic group actions. Our main result shows that a quadratic number of queries to a randomized oracle outputting LIP instances sharing the same secret is enough for inverting the group action in polynomial time. We use this result to uncover a family of weak isomorphisms and to derive two new hard problems on quadratic forms equivalent to LIP for the case of lattices with trivial automorphism.

Keywords: Group Actions · Lattice-based Cryptography · Lattice Isomorphism Problem · Post-Quantum Cryptography · Quadratic Forms

1 Introduction

Post-Quantum Cryptography is an active field of research in aiming to find valid substitutes for public-key cryptographic primitives that are used nowadays, but that are vulnerable to attacks by quantum computers. The National Institute of Standards and Technology (NIST), with a process started in 2017, selected already a few public-key cryptographic algorithms for standardization [34]. A new process started in 2023 aims to select quantum-resistant digital signatures, alternatives to the ones selected in the previous process [33].

Many cryptographic primitives in Post-Quantum Cryptography base their security on hard computational problems on algebraic structures, learning problems and decoding problems. An alternative and attractive research area is formed by those cryptographic primitives built upon isomorphisms, equivalences and group actions problems. The most studied of these sub-fields is probably the one of isogenies between supersingular elliptic curves [16,14,6,1,13,17,25,5,3,18]. However, lattice isomorphism [22,4], code-equivalence [11,7], isomorphism of multivariate polynomials [35], trilinear forms [38], and tensor isomorphism [29] revealed to be suitable for building remarkable primitives too, with a particular focus on proof-of-knowledge and digital signatures [10,30].

Informally, the *Lattice Isomorphism Problem* (LIP) in its search version consists of finding an isomorphism between two given isomorphic lattices. The decision version of the problem consists of deciding whether two given lattices are isomorphic or not.

Lattice isomorphisms have been studied and used initially in the cryptanalysis of early lattice based schemes such as NTRU [27]. Later on, Haviv and Regev studied the complexity of search-LIP [28]. More recently, two independent works by Bennett et al. [4] and by Ducas and van Woerden [22] 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 sizes [21].

Contribution. In this work, we formalize LIP from the point of view of group actions. First, we verify that lattice isomorphisms respect some general properties of group actions. Then, we provide a new result on the sufficient number of LIP samples sharing the same secret isomorphism U , so that one can efficiently recover such isomorphism. More precisely, we show that an adversary able to make $O(n^2)$ queries to an LIP randomized oracle \mathcal{O}_U can invert the group action in polynomial time and space and retrieve U . This result differs from other models, which are assumed secure even when the adversary can make a polynomial number of queries [1,20]. We provide a *sagemath* implementation of our algorithm that confirms our result and successfully recovers the secret isomorphism from a list of “independent” LIP samples. We use this result to uncover a family of weak isomorphisms, namely any commuting family of isomorphisms. Another consequence of our result is that, when used to build cryptographic primitives, the secret isomorphism of an LIP instance should not be reused in combination with different public keys.

Furthermore, 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 trivial automorphism group.

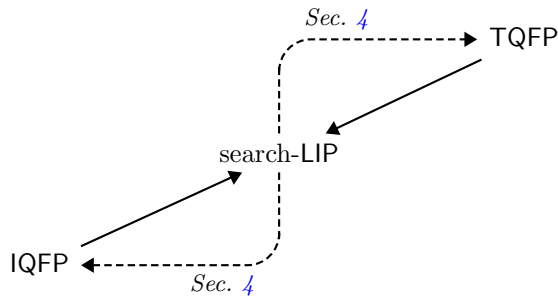


Fig. 1: Relationships between two new problems related to lattice isomorphism problem (LIP): TQFP and IQFP. Dashed arrows denote polynomial time reductions for quadratic forms with trivial automorphism group. Solid arrows denote polynomial time reductions in general.

Organization of the paper. In Section 2 we give the preliminaries on lattices and group actions. In Section 3 we formalize LIP as a group action and provide the related results. In Section 4 we introduce two new hard problems together with their reductions to LIP. Finally, in Section 5 we give our conclusions.

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., \mathbf{x}) and treat them as column vectors by default. We denote matrices by uppercase letters (e.g., M). 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 with entries in \mathbb{Z} is denoted by $\mathcal{GL}_n(\mathbb{Z}) := \{M \in \mathbb{Z}^{n \times n} : \det(M) = \pm 1\}$. For an invertible matrix $X \in \mathcal{GL}_n(\mathbb{Z})$, we denote the inverse of the transpose matrix X^\top as $X^{-\top}$. Also, by I_n we denote $n \times n$ identity matrix. The set of all *orthonormal* matrices with entries in field \mathbb{F} is denoted by $\mathcal{O}_n(\mathbb{F}) := \{O \in \mathbb{F}^{n \times n} : OO^\top = O^\top O = I_n \text{ and } \|\mathbf{o}_i\| = 1, \forall i \in \{1, \dots, n\}\}$. 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 $M_{i,j}$, namely, the matrix obtained by removing the i -th row and j -th column from M . We denote M^* the Gram-Schmidt orthogonalization of M .

A matrix $S \in \mathbb{R}^{n \times n}$ is called *symmetric positive definite* if $S = S^\top$ and $\mathbf{x}^\top S \mathbf{x} > 0$ 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 $Q = \{Q_{ij}\} \in \mathcal{S}_n^{>0}$, define $\text{unroll} : \mathcal{S}_n^{>0} \rightarrow \mathbb{R}^d$ as

$$\text{unroll}(Q) := [Q_{1,1} \ 2Q_{1,2} \ \dots \ 2Q_{1,n} \ Q_{2,2} \ 2Q_{2,3} \ \dots \ 2Q_{2,n} \ \dots \ Q_{n,n}].$$

For simplicity, in the remainder of the paper, we assume both matrix multiplication and inversion take $O(n^3)$ integer operations¹.

2.2 Lattice Isomorphisms and Quadratic Forms

We refer the reader to [22] for a more detailed introduction on the Lattice Isomorphism Problem.

A full-rank n -dimensional lattice $\mathcal{L} = \mathcal{L}(B) := B \cdot \mathbb{Z}^n$ is generated by taking all 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 $\exists U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B' = BU$. Two lattices \mathcal{L} , \mathcal{L}' are *isomorphic* if there exists an orthonormal transformation $O \in \mathcal{O}_n(\mathbb{R})$ such that $\mathcal{L}' = O \cdot \mathcal{L}$.

¹ There is a better algorithm for large dimensional matrix multiplications, the Strassen's algorithm with a running time of $O(n^{\log_2(7)})$ operations.

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

The above problem can be rephrased as follows. Given the bases $B, B' \in \mathcal{GL}_n(\mathbb{R})$ for \mathcal{L} and \mathcal{L}' respectively, find $O \in \mathcal{O}_n(\mathbb{R})$ along with $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B' = OBU$. In practice, the real-valued entries of basis 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 of B , a.k.a Gram matrix $Q := B^T B$. 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 $\mathcal{L}, \mathcal{L}'$ isomorphic lattices with respective basis B, B' , we have that $B' = OBU$ where $O \in \mathcal{O}_n(\mathbb{R})$ is orthonormal and $U \in \mathcal{GL}_n(\mathbb{Z})$ is unimodular, then we have,

$$Q' := B'^T B' = U^T B^T O^T O B U = U^T B^T B U = U^T Q U$$

where, $Q := B^T B$ is the quadratic form of B . We call Q, Q' equivalent if such $U \in \mathcal{GL}_n(\mathbb{Z})$ exists. We also denote the equivalence class by $[Q]$ of all 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 $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q' = U^T Q U$.*

The norm of vector \mathbf{x} with respect to a quadratic form Q is defined as $\|\mathbf{x}\|_Q^2 := \mathbf{x}^T Q \mathbf{x}$ and the inner product as $\langle \mathbf{x}, \mathbf{y} \rangle_Q := \mathbf{x}^T Q \mathbf{y}$. The i -th minimal distance $\lambda_i(Q)$ is defined as the smallest $r > 0$ such that $\{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{x}\|_Q \leq r\}$ 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 $Q = B_Q^T B_Q$.

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 $\text{Aut}(Q) = \{V \in \mathcal{GL}_n(\mathbb{Z}) : Q = V^T Q V\}$. We say that Q is automorphism-free if it has trivial automorphism group $\text{Aut}(Q) = \{\pm I_n\}$.*

Remark 1. Let $Q' \in [Q]$, and let $U \in \mathcal{GL}_n(\mathbb{Z})$ be such that $Q' = U^T Q U$. The set of isomorphisms between Q and Q' can be written as $\{VU : V \in \text{Aut}(Q)\}$. In other words, the automorphism group of Q determines the number of isomorphisms from Q to Q' . Equivalently, the automorphism group of Q' determines the number of isomorphisms from Q' to Q . Therefore, when Q and Q' are isomorphic, they have the same number of automorphisms. Hence, automorphism-free quadratic forms are isomorphic only to automorphism-free quadratic forms. More precisely, we have $\text{Aut}(Q') = \{\pm I_n\}$ for each quadratic form $Q' \in [Q]$.

Definition 4 (Integer Matrix Similarity Problem (IMSP)). *Given two integer matrices $A, B \in \mathbb{Z}^{n \times n}$, determine whether there exists an invertible matrix $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B = UAU^{-1}$, and if so, find U .*

The Integer Matrix Similarity Problem (also known as the Integral Conjugacy Problem) is not computationally hard. There exists indeed a probabilistic polynomial time algorithm that solves it [32,24,9].

2.3 Sampling Quadratic Forms and Unimodular Matrices

Definition 5 (Discrete Gaussian Distribution w.r.t. Quadratic Forms [22, Sec. 2.3]). For a quadratic form $Q \in \mathcal{S}_n^{>0}$, the Gaussian function on \mathbb{R}^n with parameter $s > 0$ and center \mathbf{c} is defined by

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

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

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

Brakerski *et al.* [12, Lemma 2.3] showed how to sample from a discrete Gaussian distribution efficiently. Ducas and van Woerden provide a polynomial time algorithm **Extract** that, on input a set of n linearly independent vectors Y and a quadratic form Q , returns a pair (Q', U) such that $Q' = U^T Q U$ [22, Lemma 3.1].

Definition 6 (Gaussian form distribution, [22, 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 parameter $s > 0$ is defined algorithmically as follows:

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

Ducas and van Woerden 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 $Q' = U^T Q U$, and show that $Q' \leftarrow \mathcal{D}_s([Q])$ is independent from the input class representative Q [22, 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 [8,31].

Algorithm 1 is a modified version of [8, Algorithm 4] for sampling unimodular matrices in polynomial time having the entries of the first $n - 1$ columns 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.

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

- 1: Set a matrix $M = \{M_{i,j}\} \in \mathbb{Z}^{n \times n}$ to zero
- 2: **repeat**
- 3: Sample $M_{i,j} \leftarrow [-T, T]$ uniformly at random for each $i \leq n$ and $j \leq n-1$
- 4: Use the Extended Euclidean Algorithm for computing

$$d \leftarrow \gcd\left((-1)^{n+1} \det\left(\bar{M}^{(1,n)}\right), \dots, (-1)^{2n} \det\left(\bar{M}^{(n,n)}\right)\right),$$

along with the corresponding Bézout coefficients $M_{1,j}$'s such that

$$d \leftarrow \sum_{j=1}^n M_{j,n} \cdot (-1)^{n+j} \det\left(\bar{M}^{(j,n)}\right) = \det(M)$$

- 5: **until** $d = 1$
- 6: Choose the sign of $\det(M)$ uniformly at random
- 7: Use least-squares to find the linear combination $\sum_{j=1}^{n-1} c_j [M_{1,j} \dots M_{n,j}]$ closest to $[M_{1,n} \dots M_{n,n}]$, and let \tilde{c}_i denote the nearest integer to c_i
- 8: Update $[M_{1,n} \dots M_{n,n}]$ as

$$[M_{1,n} \dots M_{n,n}] - \sum_{j=1}^{n-1} \tilde{c}_j [M_{1,j} \dots M_{n,j}]$$

- 9: **Return** M
-

2.4 Cryptographic Group Actions

We give here a refined version of some definitions on group actions introduced in [16] and [1].

Definition 7 (One-Way Function). Let P , X and Y be 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 efficient computable functions $\{f_{\text{pp}}(\cdot): X \rightarrow Y\}_{\text{pp} \in P}$ such that for all PPT adversaries \mathcal{A} we have

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

where $\text{pp} \leftarrow \mathcal{D}_P$ and $x \leftarrow \mathcal{D}_X$. If \mathcal{D}_P and \mathcal{D}_X are uniform distributions, then we simply speak of an OWF family.

Definition 8 (Weak Unpredictable Permutation). Let K and X be sets indexed by 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 $F_k^{\mathbb{S}}$ 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 $\{F(k, \cdot): X \rightarrow X\}_{k \in K}$ such that for all PPT adversaries \mathcal{A} able to query $F_k^{\mathbb{S}}$ at most t times, we have

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

where $k \leftarrow \mathcal{D}_K$ and $x^* \leftarrow \mathcal{D}_X$. If \mathcal{D}_K and \mathcal{D}_X are uniform distributions, then we simply speak of a t -wUP family.

Definition 9 (Weak Pseudorandom Permutation). Let K and X be sets indexed by 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 $\pi^{\mathbb{S}}$ a randomized oracle that samples $x \leftarrow \mathcal{D}_X$ and outputs $(x, \pi(x))$. A $(\mathcal{D}_K, \mathcal{D}_X, t)$ -weak PRP (wPRP) is a family of efficiently computable permutations $\{F(k, \cdot): X \rightarrow X\}_{k \in K}$ such that for all PPT adversaries \mathcal{A} able to query $\pi^{\mathbb{S}}$ at most t times, we have

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

where $k \leftarrow \mathcal{D}_K$, $\pi \leftarrow S_X$. If \mathcal{K} and \mathcal{D} are uniform distributions, then we simply speak of a t -wPRP family.

Definition 8 and Definition 9 give more fine-grained notions in comparison to their respective in [1, Section 2.1]. In particular, our definitions include a limit on the number of queries that an adversary can make to the oracle. A similar setting can be found in [15,37,20].

Definition 10 (Group Action). A group (G, \circ) is said to act on a set X if there is a map $\star: G \times X \rightarrow X$ 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 $\pi_g: x \mapsto g \star x$ defines a permutation of X .

Definition 11 (Properties of Group Actions).

1. A group action (G, X, \star) is said to be **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. A group action (G, X, \star) is said to be **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. A group action (G, X, \star) is said to be **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 $x = g \star x$.

4. A group action (G, X, \star) is said to be **regular** if it is both free and transitive. For such a regular group action, the set X is called a principal homogeneous space for the group G , or a G -torsor.

Definition 12 (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 $\{f_x: G \rightarrow X\}_{x \in X}$ is $(\mathcal{D}_G, \mathcal{D}_X)$ -one-way, where $f_x: g \mapsto g \star x$, and $\mathcal{D}_X, \mathcal{D}_G$ are distributions on X, G respectively.

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

Definition 14 (Weak Pseudorandom Group Action). A group action (G, X, \star) is $(\mathcal{D}_X, \mathcal{D}_G, t)$ -weakly pseudorandom if the family of efficiently computable permutations $\{\pi_g: X \rightarrow X\}_{g \in G}$ is a $(\mathcal{D}_X, \mathcal{D}_G, t)$ -weak PRP where π_g is defined as $\pi_g: x \mapsto g \star x$ and $\mathcal{D}_X, \mathcal{D}_G$ are distributions on X, G respectively.

3 Lattice Isomorphism as a Group Action

In this section we introduce lattice isomorphisms, in the quadratic form version, as a group action, and we provide some results related to it. Consider the equivalence relation \simeq_{\pm} defined as

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

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

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

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

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

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

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

$$\star: (\mathcal{GL}_n^\pm(\mathbb{Z}) \times [Q]) \rightarrow [Q], \quad \star(V, Q_0) \mapsto V \star Q_0 := V^\top Q_0 V,$$

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

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

$$(U \cdot V) \star Q_0 = (VU)^\top Q_0 VU = U^\top (V^\top Q_0 V) U = U \star (V^\top Q_0 V) = U \star (V \star Q_0),$$

which proves the compatibility. \square

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 [1,30,11], in our case we have that both the base set and the group are infinite.

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 $(\mathcal{GL}_n^\pm(\mathbb{Z}), [Q], \star)$ is transitive and faithful.

Proof. We begin by proving the transitivity. If $Q_0, Q_1 \in [Q]$, then Q_0 and Q_1 are isomorphic to Q , that is, there exist $U, V \in \mathcal{GL}_n^\pm(\mathbb{Z})$ such that $Q_0 = U^\top Q U$ and $Q_1 = V^\top Q V$. Then, one has that $Q_1 = (U^{-1}V)^\top Q_0 U^{-1}V$ and $U^{-1}V \in \mathcal{GL}_n^\pm(\mathbb{Z})$ maps Q_0 to Q_1 via the group action \star . This proves the transitivity property and, hence, the equivalence class $[Q]$ is a homogeneous space for $\mathcal{GL}_n^\pm(\mathbb{Z})$.

We prove now the group action to be faithful by contradiction. Let $U \neq I_n \in \mathcal{GL}_n^\pm(\mathbb{Z})$ and assume that fixes every element of $[Q]$. Then for every $Q_0 \in [Q]$, we have that $U \star Q_0 = Q_0$. Let $V \in \mathcal{GL}_n^\pm(\mathbb{Z})$ any unimodular different from the identity and let $Q_1 = V \star Q_0$. Since $Q_1 \in [Q]$, we have that

$$(V \cdot U) \star Q_0 = U \star Q_1 = Q_1 = V \star Q_0 = V \star Q_0 = V \star (U \star Q_0) = (U \cdot V) \star Q_0.$$

In other words, for every $Q_0 \in [Q]$ and every $V \in \mathcal{GL}_n^\pm(\mathbb{Z})$, U and V always commute in the group operation of $\mathcal{GL}_n^\pm(\mathbb{Z})$. This, however happens only for $U = I_n$. \square

The following proposition sets a condition for the **free** condition to be satisfied.

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

Proof. In this proof, in order to avoid confusion, we bring back the equivalence class notation for the elements of $\mathcal{GL}_n^\pm(\mathbb{Z})$. Assume Q to be automorphism-free. Then, for $Q_0 \in [Q]$, if $V^\top Q_0 V = Q_0$, we have that $V \in \mathcal{GL}_n(\mathbb{Z})$ is an

automorphism for Q and therefore $V = \pm I_n$, that is, $V \in [I_n]_{\pm} \in \mathcal{GL}_n^{\pm}(\mathbb{Z})$. On the contrary, if for every given quadratic form $Q_0 \in [Q]$, $[I_n]_{\pm}$ is the only element of $\mathcal{GL}_n^{\pm}(\mathbb{Z})$ that fixes Q_0 , then Q_0 has only trivial automorphisms (I_n and $-I_n$) as well as every element of the class $[Q]$. Therefore, Q is automorphism-free. \square

Theorem 1 introduces a new result for LIP that gives a sufficient number of oracle queries for an adversary to invert the group action in polynomial time and space. Given the generality of the result, we do not limit on any specific distribution on the group $\mathcal{GL}_n^{\pm}(\mathbb{Z})$ for the secret unimodular matrix. On the contrary, concerning the distribution on the base set $[Q]$, we need the distribution to satisfy the following property.

Definition 15. 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 $\mathcal{D}_{[Q]}$ is p -linearly strong if, given $Q_1, \dots, Q_p \leftarrow \mathcal{D}_{[Q]}$, the $p \times d$ matrix $M_{\mathcal{Q}}$ whose rows are $\text{unroll}(Q_i)$ (see definition in Section 2) is such that

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

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

Proof. We show that the $(\mathcal{GL}_n^{\pm}(\mathbb{Z}), [Q], \star)$ is not a $(\mathcal{D}_{[Q]}, \mathcal{D}_{\mathcal{GL}_n^{\pm}(\mathbb{Z})}, d)$ -weak unpredictable group action by providing a polynomial-time algorithm **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 $(Q, V^{\top} Q V) \in \mathcal{S}_n^{>0} \times \mathcal{S}_n^{>0}$. Then, the adversary \mathcal{A} is able to collect a list of d pairs $\mathcal{Q} := \{(Q_i, V^{\top} Q_i V)\}_{i=1, \dots, d}$ such that the $d \times d$ matrix $M_{\mathcal{Q}}$ whose rows are composed by $\text{unroll}(Q_i)$ is full rank with probability $1 - \text{negl}(n)$.

We describe first a procedure **Linearize**, sub-routine of the main algorithm **Recover** to compute the secret unimodular V . The underlying idea takes inspiration from the work of Rasslan and Youssef [36].

Procedure Linearize. Consider one pair $(Q, Q' = V^{\top} Q V)$ 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

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

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

$$\mathbf{Q} = [Q_{1,1} \ 2Q_{1,2} \ \dots \ 2Q_{1,n} \ Q_{2,2} \ 2Q_{2,3} \ \dots \ 2Q_{2,n} \ \dots \ Q_{n,n}], \text{ and}$$

$$\mathbf{x}_i = [X_{(i,1),(i,1)} \ \dots \ X_{(i,1),(i,n)} \ X_{(i,2),(i,2)} \ \dots \ X_{(i,2),(i,n)} \ \dots \ X_{(i,n),(i,n)}]^\top.$$

For $i \neq j$, we rewrite Equation (2) as follows:

$$Q'_{ij} = \sum_{k=1}^n \sum_{l=k+1}^n 2Q_{k,l} \cdot \underbrace{\left(\frac{X_{(i,k),(j,l)} + X_{(i,l),(j,k)}}{2} \right)}_{Y_{(i,k),(j,l)}} + \sum_{k=1}^n Q_{k,k} \cdot X_{(i,k),(j,k)}. \quad (3)$$

Let $\mathbf{y}_{i,j}$ be the d -dimensional vector with coefficients $Y_{(i,k),(j,l)}$ and $X_{(i,k),(j,k)}$ given by

$$\mathbf{y}_{i,j} = [X_{(i,1),(j,1)} \ Y_{(i,1),(j,2)} \ \dots \ Y_{(i,1),(j,n)} \ X_{(i,2),(j,2)} \ Y_{(i,1),(j,3)} \ \dots \ X_{(i,n),(i,n)}]^\top.$$

Then we have that $Q'_{i,j} = \mathbf{Q} \cdot \mathbf{y}_{i,j}$ and

$$\mathbf{Q}' = \mathbf{Q} \cdot \overbrace{[\mathbf{x}_1 \ \mathbf{y}_{1,2} \ \dots \ \mathbf{y}_{1,n} \ \mathbf{x}_2 \ \mathbf{y}_{2,3} \ \dots \ \mathbf{y}_{2,n} \ \dots \ \mathbf{x}_n]}^{d\text{-by-}d \text{ matrix}} \quad (4)$$

$\downarrow \quad \downarrow$
 $d\text{-dimensional vectors}$

where

$$\mathbf{Q}' = [Q'_{1,1} \ Q'_{1,2} \ \dots \ Q'_{1,n} \ Q'_{2,2} \ Q'_{2,3} \ \dots \ Q'_{2,n} \ \dots \ Q'_{n,n}].$$

Algorithm Recover. The procedure `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 to derive d^2 linearly independent equations and, therefore, proceed by finding the unique solution to the associated system. We describe the algorithm to recover V below and we will refer to it as `Recover`:

1. For each pair (Q_i, Q'_i) in \mathcal{Q} , apply `Linearize`(Q_i, Q'_i) and get the following equation

$$\mathbf{Q}'_i = \mathbf{Q}_i \cdot [\mathbf{x}_1 \ \mathbf{y}_{1,2} \ \dots \ \mathbf{y}_{1,n} \ \mathbf{x}_2 \ \mathbf{y}_{2,3} \ \dots \ \mathbf{y}_{2,n} \ \dots \ \mathbf{x}_n].$$

2. Solve the linear system

$$\begin{bmatrix} \mathbf{Q}'_1 \\ \vdots \\ \mathbf{Q}'_d \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 \\ \vdots \\ \mathbf{Q}_d \end{bmatrix} \cdot [\mathbf{x}_1 \ \mathbf{y}_{1,2} \ \dots \ \mathbf{y}_{1,n} \ \mathbf{x}_2 \ \mathbf{y}_{2,3} \ \dots \ \mathbf{y}_{2,n} \ \dots \ \mathbf{x}_n].$$

to retrieve $\mathbf{x}_1, \dots, \mathbf{x}_n$ as follows

$$\mathbf{z} = [\mathbf{x}_1 \ \mathbf{y}_{1,2} \ \dots \ \mathbf{y}_{1,n} \ \mathbf{x}_2 \ \mathbf{y}_{2,3} \ \dots \ \mathbf{y}_{2,n} \ \dots \ \mathbf{x}_n] = \begin{bmatrix} \mathbf{Q}_1 \\ \vdots \\ \mathbf{Q}_d \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q}'_1 \\ \vdots \\ \mathbf{Q}'_d \end{bmatrix}.$$

By construction, solution \mathbf{z} has rational values concerning the entries $Y_{(i,k),(j,l)}$ from $\mathbf{y}_{i,j}$. In other entries different from $Y_{(i,k),(j,l)}$, the values are the integers determined by $X_{(i,k),(j,l)}$.

3. Derive the entries of the solution matrix U determined by \mathbf{z} by computing first $U_{1,1} = \sqrt{\mathbf{z}_{1,1}}$, then $U_{j,1} = \frac{\mathbf{z}_{j,1}}{U_{1,1}}$ for $j \leq n$, and so on for each single entry in \mathbf{z} . More precisely, we have $U_{i,1} = \frac{\mathbf{z}_{i,1}}{U_{1,1}}$ for each $i \leq n$, and $U_{i,j} = \frac{\mathbf{z}_{i,j}}{U_{i,1}}$, where $k = \sum_{l=1}^{i-1} (n-l+1)$ for each $j = 1, \dots, n$. We have the following two scenarios:
 - (a) If $U_{i,1} \neq 0$ for $i = 1, \dots, n$ then $U = \pm V$ and the algorithm terminates.
 - (b) If $U_{i,1} = 0$ for some $1 \leq i \leq n$, then the algorithm cannot recover the full matrix U as there would be a division by zero. In this case, one samples a unimodular matrix R using Algorithm 1 for a parameter $T = O(n)$. So, one computes the set $\mathcal{Q}' := \{(Q, R^T Q' R) : (Q, Q') \in \mathcal{Q}\}$ and repeats **Recover** with \mathcal{Q}' as input. Note that $M_{\mathcal{Q}'} = M_{\mathcal{Q}}$, and so $\text{rank}(M_{\mathcal{Q}'}) = d$. If one succeeds at recovering the matrix $W = UR$ (i.e., W has only non-zero entries in its first column). Then one recovers $\pm V$ as $U = WR^{-1}$ and the algorithm terminates. Otherwise, one tries again with a different unimodular matrix R until it succeeds.

Memory and time complexities. **Recover** requires one d -dimensional matrix inversion and one d -dimensional matrix multiplication. The last step of deriving the entries of V takes $O(n^2)$ integer operations. Recall that $d = \frac{n(n+1)}{2}$. Then the time complexity of deriving V becomes $O\left(\frac{n^3(n+1)^3}{4} + n^2\right)$ operations. Since we need to store four d -dimensional matrices, we have a memory complexity equal to $O(4d^2) = O(n^2(n+1)^2)$.

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

$$V_{j,1}x_1 + V_{j,2}x_2 + \dots + V_{j,n}x_n = 0, \quad \text{for some } 1 \leq j \leq n. \quad (5)$$

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

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

that is, x_n is uniquely determined by x_1, \dots, x_{n-1} and, whether or not (x_1, \dots, x_{n-1}) leads or not to a solution is determined by a congruence condition modulo $V_{j,n}$.

Thus, for every j , there exists a rational constant $0 \geq \gamma_j \geq 1$ 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 $[R_{1,1} \dots R_{n,1}]$ is not a solution of any of Equation (5) is at least

$$\left(1 - \frac{1}{2T + 1}\right)^n = \left(1 - \frac{1}{O(n)}\right)^n = \left(1 - \frac{1}{cn}\right)^n \xrightarrow{n \rightarrow \infty} e^{-1/c}, \quad \text{for some } c \geq 1.$$

□

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

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

Theorem 1 and Corollary 1 can be easily generalized to the case of a $\mathcal{D}_{[Q]}$ being p -linearly strong, for $p > d$, when the adversary is able to make p or more queries to the random oracle.

On d -Linearly Strong Distributions and Experimental Verification. We believe that the hypothesis on the distribution $\mathcal{D}_{[Q]}$ to be d -linearly strong is realistic. Essentially, we require $\mathcal{D}_{[Q]}$ to output quadratic forms that are linearly independent from each other via the function `unroll()`. On the other hand, a distribution that outputs samples that are somewhat linearly dependant would likely come with serious security implications when used to build cryptographic primitives. We could not prove that $\mathcal{D}_s([Q])$ (described in Definition 6, introduced and used in [22]) is d -linearly strong theoretically. However, we heuristically verified that $\mathcal{D}_s([Q])$ behaves as a d -linearly strong distribution. On the basis of this, we consider the following assumption for what comes in Section 4.

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

Using $\mathcal{D}_s([Q])$ as distribution for the base set $[Q]$ and several different distributions on $\mathcal{GL}_n^\pm(\mathbb{Z})$, we verified the correctness of `Recover` presented in the proof of Theorem 1 via a `sagemath` implementation available as an attachment to this manuscript.

3.1 Weak subgroups of $\mathcal{GL}(\mathbb{Z})$

Theorem 1 shows that it is enough to have $\frac{n(n+1)}{2}$ “independent” LIP instances sharing the same secret unimodular V to allow an efficient recovery of it. In this section, we use this fact to uncover a new family of weak LIP instances. Namely, if the secret unimodular matrix belongs to a commutative subgroup of

$\mathcal{GL}(\mathbb{Z})$ (eg. circulant matrices, powers of a matrix, ...), then it can be recovered in polynomial time.

Let $\mathcal{G}_c \subset \mathcal{GL}(\mathbb{Z})$ be a commutative group, and let $V \in \mathcal{G}_c$. Given an LIP instance $(Q, Q' = V^\top Q V)$, 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 a unimodular matrix $U \in \mathcal{G}_c$ and compute

$$(\bar{Q} := U^\top Q U, \bar{Q}' := U^\top Q' U = U^\top V^\top Q V U = V^\top U^\top Q U V = V^\top \bar{Q} V).$$

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 `Recover` described in the proof of Theorem 1 to retrieve it.

4 New Hard Problems on Quadratic Forms

Another implication of Theorem 1 is the fact that it implicitly introduces two new LIP-equivalent computational hard problems on quadratic forms. This section introduces those hinted two new problems and provides their polynomial-time reductions to sLIP_Q , when Q is automorphism-free.

Definition 16 (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 $Q = B^\top B$. Given $Q' \in [Q]$, the Transpose Quadratic Form Problem is to compute $\hat{Q} \in [Q]$ such that $\hat{Q} = U Q U^\top$, where $U \in \mathcal{GL}_n(\mathbb{Z})$ satisfies $Q' = U^\top Q U$.*

Definition 17 (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 $Q = B^\top B$. Given $Q' \in [Q]$, the Inverse Quadratic Form Problem is to compute $\hat{Q} \in [Q]$ such that $\hat{Q} = U^{-\top} Q U^{-1}$, where $U \in \mathcal{GL}_n(\mathbb{Z})$ satisfies $Q' = U^\top Q U$.*

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 $S_{Q'} := \{\tilde{Q}_V = (VU)Q(VU)^\top : V \in \text{Aut}(Q)\}$. For the specific case of automorphism-free quadratic forms, the solution is unique ($|S_{Q'}| = 1$). Taking Assumption 1 for true, we give in Lemma 1 and Lemma 2 polynomial-time reductions from sLIP_Q to TQFP and IQFP respectively. We implemented and successfully tested these reductions in a `sagemath` script available as an attachment to this manuscript.

Lemma 1 (From sLIP_Q to TQFP). *Let $Q \in \mathcal{S}_n^{>0}$ be an automorphism-free quadratic form. Given an oracle $\mathcal{O}_{\text{TQFP}}$ that solves TQFP in time T_0 , there is an algorithm that solves sLIP_Q in expected time $O(n^2(T_0 + T_1) + n^6)$, where T_1 is the time complexity of one call to $\mathcal{D}_s([Q])$, for $s \geq \max\{\lambda_n(Q), \|B_Q^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}$.*

Proof. Let us fix the same setup as Definition 16, then we have Q and $Q' = V^\top QV$, where $Q' \in [Q]$. For simplicity, we assume that $\mathcal{O}_{\text{TQFP}}$ always solves TQFP for isomorphic input Q, Q' . We give an algorithm which solves sLIP_Q with a polynomial number of calls to $\mathcal{O}_{\text{TQFP}}$ as follows. Let us set $d = \frac{n(n+1)}{2}$.

1. Forward (Q', Q) to $\mathcal{O}_{\text{TQFP}}$ and receive the response $\widehat{Q} = VQV^\top$.
2. (a) Sample a quadratic form $\bar{Q} = W^\top QW$ along with $W \in \mathcal{GL}_n(\mathbb{Z})$ from $\mathcal{D}_s([Q])$.
 (b) Compute $Q'' = W\widehat{Q}W^\top = WVQV^\top W^\top$ and send (Q'', Q) to $\mathcal{O}_{\text{TQFP}}$. Record its response as $\bar{Q} = V^\top W^\top QWV = V^\top QV$.
 (c) Compute $Q''' = WQ'W^\top = WV^\top QVW^\top$ and send (Q''', Q) to $\mathcal{O}_{\text{TQFP}}$. Record its response as $\widehat{Q} = VW^\top QWV^\top = V\bar{Q}V^\top$.
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 $\text{unroll}\left(Q_0^{(i)}\right)$ is full rank.
4. Retrieve $V \leftarrow \text{Recover}(\mathcal{Q})$ as described in Theorem 1.

Running time. Let us assume both matrix multiplication and inversion take $O(n^3)$ integer operations. Step 1 costs one call to the oracle $\mathcal{O}_{\text{TQFP}}$. Step 2 samples one random unimodular matrix, makes four matrix multiplications, and two queries to $\mathcal{O}_{\text{TQFP}}$. 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 has a complexity equals to

$$O\left(T_0 + \frac{n(n+1)}{2} (2T_0 + T_1 + 4n^3)\right) = O(n^2(T_0 + T_1) + n^5).$$

Step 4 requires $O(n^6)$ operations to retrieve V , which gives a total asymptotic time complexity of $O(n^2(T_0 + T_1) + n^6)$. \square

Remark 2. Regarding Lemma 1, in practice one can reduce the number of calls to $\mathcal{O}_{\text{TQFP}}$ by a factor of n by exploiting the following. Let $Q, Q', \widehat{Q} \in \mathcal{S}_n^{>0}$ be equivalent quadratic forms with $Q' = V^\top QV$ and $\widehat{Q} = VQV^\top$, for some unimodular matrix $V \in \mathcal{GL}_n(\mathbb{Z})$. Then, one can compute the quadratic forms

$$Q_1 := Q'QQ' = V^\top QVQV^\top QV, \quad Q_0 := Q\widehat{Q}Q,$$

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

$$Q_1^{(i)} := Q'(QQ')^i, \quad Q_0^{(i)} := Q(\widehat{Q}Q)^i,$$

with $Q_1^{(i)} = V^\top Q_0^{(i)}V$, 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 $M^n \in \text{Span}\{I_n, M, M^2, \dots, M^{n-1}\}$ [2, §7.11]. Therefore, with the above approach, we can get a set $\mathcal{Q} = \{(Q_i, Q'_i = V^\top Q_iV)\}_{i=1}^p$ of size $p \leq n$ from the

knowledge of $Q' = V^T Q V$ and $\widehat{Q} = V Q V^T$. Using this trick in Step 2 of the proof of Lemma 1, and assuming that p reaches n with high probability, one can reduce the number of calls to $\mathcal{O}_{\text{TQFP}}$ 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 1 would be $\widetilde{O}(n(T_0 + T_1) + n^6)$.² In our *sagemath* implementation, we implemented and tested the variant of the reduction in Lemma 1 that uses such optimization in Step 2.

Lemma 2 (From sLIP_Q to IQFP). *Let $Q \in \mathcal{S}_n^{>0}$ be an automorphism-free quadratic form. Given an oracle $\mathcal{O}_{\text{IQFP}}$ that solves IQFP in time T_0 , there exists an algorithm that solves sLIP_Q in expected time $O(n^2(T_0 + T_1) + n^6)$, where T_1 is the time complexity of one call to $\mathcal{D}_s([Q])$, for $s \geq \max\{\lambda_n(Q), \|B_Q^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}$.*

Proof. Let us fix the same setup as Definition 17, then we have Q and $Q' = V^T Q V$, where $Q' \in [Q]$. For simplicity, we assume that $\mathcal{O}_{\text{IQFP}}$ always solves IQFP, for a isomorphic input Q, Q' . We give an algorithm which solves sLIP_Q with a polynomial number of calls to $\mathcal{O}_{\text{IQFP}}$ as follows. Let us set $d = \frac{n(n+1)}{2}$.

1. Forward (Q', Q) to $\mathcal{O}_{\text{IQFP}}$ and receive the response $\widehat{Q} = V^{-T} Q V^{-1}$.
2. (a) Sample a quadratic form $\bar{Q} = W^T Q W$ along with $W \in \mathcal{GL}_n(\mathbb{Z})$ from $\mathcal{D}_s([Q])$.
 (b) Calculate $Z = W^{-1}$.
 (c) Compute $Q'' = Z^T \widehat{Q} Z = Z^T V Q V^T Z$ and send (Q'', Q) to $\mathcal{O}_{\text{IQFP}}$. Record its response as $\widetilde{Q} = V^T W^T Q W V = V^T \bar{Q} V$.
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 $\text{unroll}\left(Q_0^{(i)}\right)$ is full rank.
4. Retrieve $V \leftarrow \text{Recover}(\mathcal{Q})$ as described in Theorem 1.

Running time. The cost analysis is analogous to Lemma 1, with the addition of a matrix inversion in Step 2. However, this is negligible on the total cost of the reduction, that is $O(n^2(T_0 + T_1) + n^6)$. □

To illustrate the above reductions from Lemma 1 and Lemma 2, we simulate the algorithms concerning TQFP and IQFP using a *sagemath* library [39]; we provide our code as supplementary material.

Remark 3. An adversary having access to both $\mathcal{O}_{\text{TQFP}}$ and $\mathcal{O}_{\text{IQFP}}$ can solve sLIP_Q with only one query to each of the two oracles in polynomial time. Indeed, let $Q' = U^T Q U$. Query $\mathcal{O}_{\text{TQFP}}$ and $\mathcal{O}_{\text{IQFP}}$ to obtain

$$Q_1 = U Q U^T \quad \text{and} \quad Q_2 = U^{-T} Q U^{-1}.$$

² We have $\widetilde{O}(\cdot)$ instead of $O(\cdot)$ because of the increase of the integer coefficients size when applying this optimization trick.

The product of these gives

$$Q_1Q_2 = UQ^2U^{-1}.$$

Now, the key observation here is that U can be retrieved by solving an IMSP instance (see Definition 4) with $A = Q^2$ and $B = Q_1Q_2$ as input for which there exists a probabilistic polynomial time [9,26]. Since we assume Q has trivial automorphism, the algorithm is expected to output $\pm U$.

Remark 4. Lemma 1 and Lemma 2 can be generalized to the case of lattices 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 $\mathcal{O}_{\text{TQFP}}$ that returns one of the possible solutions uniformly at random. Then, the algorithm in Lemma 1 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 $\mathcal{O}_{\text{TQFP}}$, one must repeat on average the whole algorithm $(|\text{Aut}(Q)|/2)^n$ times.

5 Conclusions and Future Directions

In this work we formalized lattice isomorphism as a group action and proved some properties of it. We introduced a result that gives the sufficient number of instances sharing the same secret for the problem to be solvable in polynomial time. The consequences of our work include a new family of weak isomorphisms and the fact that secret-keys in this context must not be reused in combinations with other public keys.

As a future work, it would be interesting to investigate whether an analogous result can be obtained also for other group actions and equivalence problems (e.g. code equivalence). More in general, it would be interesting to investigate whether other group actions also come with a similar limitation on the number of queries allowed to an adversary. We introduced two new hard problems and applied our result to prove them to be equivalent to sLIP for the case of quadratic forms with trivial automorphism group. We also leave as a future work to investigate the possible applications of these in building new cryptographic primitives.

Acknowledgments The authors thank Keita Xagawa and Victor Mateu for fruitful discussions on the topic. We also thank Elena Kirshanova and anonymous reviewers for the useful comments on a earlier version of this manuscript.

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. Banegas, G., Bernstein, D.J., Campos, F., Chou, T., Lange, T., Meyer, M., Smith, B., Sotáková, J.: CTIDH: faster constant-time CSIDH. IACR TCHES **2021**(4), 351–387 (2021). <https://doi.org/10.46586/tches.v2021.i4.351-387>, <https://tches.iacr.org/index.php/TCHES/article/view/9069>
4. Bennett, H., Ganju, A., Peetathawatchai, P., Stephens-Davidowitz, N.: Just how hard are rotations of zn ? algorithms and cryptography with the simplest lattice. In: Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23–27, 2023, Proceedings, Part V. p. 252–281. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-30589-4_9, https://doi.org/10.1007/978-3-031-30589-4_9
5. Beullens, W., Disson, L., Pedersen, R., Vercauteren, F.: CSI-RAShI: Distributed key generation for CSIDH. In: Cheon, J.H., Tillich, J.P. (eds.) Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021. pp. 257–276. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-030-81293-5_14
6. 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
7. 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
8. 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. pp. 319–338. Springer International Publishing, Cham (2021)
9. Bley, W., Hofmann, T., Johnston, H.: Computation of lattice isomorphisms and the integral matrix similarity problem. Forum of Mathematics, Sigma **10**, e87 1–36 (2022). <https://doi.org/10.1017/fms.2022.74>, <https://doi.org/10.1017/fms.2022.74>
10. 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>, <https://eprint.iacr.org/2022/1184>
11. 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>, <https://eprint.iacr.org/2023/718>
12. 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>
13. Castryck, W., Decru, T.: CSIDH on the surface. In: Ding and Tillich [19], pp. 111–129. https://doi.org/10.1007/978-3-030-44223-1_7
14. 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

15. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_26
16. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Paper 2006/291 (2006), <https://eprint.iacr.org/2006/291>, <https://eprint.iacr.org/2006/291>
17. Cozzo, D., Smart, N.P.: Sashimi: Cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol. In: Ding and Tillich [19], pp. 169–186. https://doi.org/10.1007/978-3-030-44223-1_10
18. 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
19. Ding, J., Tillich, J.P. (eds.): Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. Springer, Heidelberg (2020)
20. Dodis, Y., Puniya, P.: Feistel networks made public, and applications. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 534–554. Springer, Heidelberg (May 2007). https://doi.org/10.1007/978-3-540-72540-4_31
21. 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
22. Ducas, L., van Woerden, W.P.J.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Dunkelman and Dziembowski [23], pp. 643–673. https://doi.org/10.1007/978-3-031-07082-2_23
23. Dunkelman, O., Dziembowski, S. (eds.): EUROCRYPT 2022, Part III, LNCS, vol. 13277. Springer, Heidelberg (May / Jun 2022)
24. Eick, B., Hofmann, T., O’Brien, E.A.: The conjugacy problem in $GL(n, \mathbb{Z})$. J. Lond. Math. Soc. **100**(3), 731–756 (2019). <https://doi.org/10.1112/jlms.12246>, <https://doi.org/10.1112/jlms.12246>
25. El Kaafarani, A., Katsumata, S., Pintore, F.: Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part II. LNCS, vol. 12111, pp. 157–186. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45388-6_6
26. Fieker, C., Hart, W., Hofmann, T., Johansson, F.: Nemo/Hecke: Computer Algebra and Number Theory Packages for the Julia Programming Language. In: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation. pp. 157–164. ISSAC ’17, ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3087604.3087611>, <https://doi.acm.org/10.1145/3087604.3087611>
27. 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
28. 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>
29. 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

30. Joux, A.: Mpc in the head for isomorphisms and group actions. Cryptology ePrint Archive, Paper 2023/664 (2023), <https://eprint.iacr.org/2023/664>, <https://eprint.iacr.org/2023/664>
31. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective, vol. 671 (01 2002). <https://doi.org/10.1007/978-1-4615-0897-7>
32. Myasnikov, A.D., Ushakov, A.: Cryptanalysis of matrix conjugation schemes. J. Math. Cryptol. **8**(2), 95–114 (2014). <https://doi.org/10.1515/jmc-2012-0033>, <https://doi.org/10.1515/jmc-2012-0033>
33. NIST: Post-quantum cryptography: Digital signature schemes. <https://csrc.nist.gov/projects/pqc-dig-sig>
34. NIST: Post-quantum cryptography standardization. <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
35. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) EU-ROCRYPT'96. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (May 1996). https://doi.org/10.1007/3-540-68339-9_4
36. 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>, <https://doi.org/10.1587/transfun.E94.A.853>
37. Sjödin, J.: Weak Pseudorandomness and Unpredictability. Ph.D. thesis, ETH Zurich (2007), eTH Series in Information Security and Cryptography, vol. 8, Hartung-Gorre Verlag, ISBN 3-86628-088-2
38. 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 [23], pp. 582–612. https://doi.org/10.1007/978-3-031-07082-2_21
39. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 9.7) (2022), <https://www.sagemath.org>