On the Lattice Isomorphism Problem, Quadratic Forms, Remarkable Lattices, and Cryptography

Léo Ducas[1,2] and Wessel van Woerden[1(✉)]

[1] CWI, Cryptology Group, Amsterdam, The Netherlands
www@cwi.nl
[2] Mathematical Institute, Leiden University, Leiden, The Netherlands

Abstract. A natural and recurring idea in the knapsack/lattice cryptography literature is to start from a lattice with remarkable decoding capability as your private key, and hide it somehow to make a public key. This is also how the code-based encryption scheme of McEliece (1978) proceeds.

This idea has never worked out very well for lattices: ad-hoc approaches have been proposed, but they have been subject to ad-hoc attacks, using tricks beyond lattice reduction algorithms. On the other hand the framework offered by the Short Integer Solution (SIS) and Learning With Errors (LWE) problems, while convenient and well founded, remains frustrating from a coding perspective: the underlying decoding algorithms are rather trivial, with poor decoding performance.

In this work, we provide generic realizations of this natural idea (independently of the chosen remarkable lattice) by basing cryptography on the lattice isomorphism problem (LIP). More specifically, we provide:

- a worst-case to average-case reduction for search-LIP and distinguish-LIP within an isomorphism class, by extending techniques of Haviv and Regev (SODA 2014).
- a zero-knowledge proof of knowledge (ZKPoK) of an isomorphism. This implies an identification scheme based on search-LIP.
- a key encapsulation mechanism (KEM) scheme and a hash-then-sign signature scheme, both based on distinguish-LIP.

The purpose of this approach is for remarkable lattices to improve the security and performance of lattice-based cryptography. For example, decoding within poly-logarithmic factor from Minkowski's bound in a remarkable lattice would lead to a KEM resisting lattice attacks down to poly-logarithmic approximation factor, provided that the dual lattice is also close to Minkowski's bound. Recent works have indeed reached such decoders for certain lattices (Chor-Rivest, Barnes-Sloan), but these do not perfectly fit our need as their duals have poor minimal distance.

1 Introduction

At repeated occasions [8,22,24,33,45], and over more than 30 years, it has been attempted to adapt the public-key encryption scheme of McEliece [25] from codes

O. Dunkelman and S. Dziembowski (Eds.): EUROCRYPT 2022, LNCS 13277, pp. 643–673, 2022.
https://doi.org/10.1007/978-3-031-07082-2_23

to lattices. More specifically, these works attempted to construct particularly good lattices with efficient decoding algorithms, to use it as a secret-key, and to give a bad description of a similar lattice as the corresponding public-key. For example, it was analysed in [11] that the Chor-Rivest cryptosystem [8] was implicitly relying on a family of lattices for which it is possible to efficiently decode errors up to a radius within a factor of $O(\log n)$ from optimal (Minkowski bound). For comparison, the decoding algorithm underlying schemes based on the Learning with Error problem [39] (LWE) fall short from the Minkowski bound by polynomial factors; they essentially reduce decoding to the case of the trivial lattice $\mathbb{Z}^n$.

This McEliece-like approach has unfortunately not been very popular lately. Perhaps it has suffered from the failure of the Merkle-Hellman Knapsack-based cryptosystem [26,43] more than it should have. Indeed, from the "knapsack-era", only the Merkle-Hellman cryptosystem and its variants were completely devastated by a polynomial-time attack [32]. In contrast, the best known attack against the scheme of Chor and Rivest [8,24] remains sub-exponential in the dimension n; what may be concerning is that those attacks are not pure lattice reduction attacks. For both versions of this scheme, the canonical coordinates are partially brute-forced during the best attack. Lapiha [20] found that an Information Set Decoding attack was possible against the variant of Li *et al.* [24]. Brickell's attack against the original scheme also relies on guessing over a few canonical coordinates, inside of an arithmetic attack [8, Sec. VII.5].

However, we note that these attacks are enabled by the fact that these schemes only re-randomize the lattice by applying a permutation of the coordinates.[1] Such permutations are isometries, i.e. lattice isomorphism, but those are not the only ones... The isometry group $\mathcal{O}_n(\mathbb{R})$ acting on lattices is much larger than the one acting on codes, and applying a random isometry from this larger group should convincingly thwart those code-style attacks: the canonical coordinate system becomes irrelevant.

All these remarks point toward the Lattice Isomorphism Problem (LIP) as a potential theoretical platform for finally getting this natural approach properly formalized, and hopefully, truely "lattice-based" in the cryptanalytic sense: the best known attack should be based on generic lattice reduction algorithms such as LLL [21] and BKZ [40]. The current state of the art on LIP supports this hypothesis: all known algorithms [17,37,38,44] rely on finding short vectors. This is the case even for algorithms specialized to the trivial lattice $\mathbb{Z}^n$ [46]. However, experimental studies [6] show that the basis randomization step requires care.

While instantiating LIP with $\mathbb{Z}^n$ may already give rise to secure cryptosystems, the end goal of this work is to enable lattice-based cryptosystems that

[1] This permutation is in fact implicit, hidden in the ordering of the evaluation points used to define the lattice. Furthermore, both in these lattice schemes and in subsequent versions of the McEliece, one may also discard some the evaluation points to randomize the lattice/code itself beyond isometry. In this article, we will not consider this extra randomization.

could be even more secure than those based on LWE and SIS, by instantiating the constructed schemes with remarkably decodable lattices.

1.1 Contributions

We propose a formal and convenient framework for LIP-based cryptography, from which we build an identification scheme based on search-LIP (sLIP), a (passively secure) Key Encapsulation Mechanism (KEM) based on distinguish-LIP (ΔLIP), as well as signature scheme also based on ΔLIP. In more details:

- We first discuss the LIP problem, recall the quadratic form formalism (Sect. 2.2), and rephrase the LIP problem in terms of quadratic forms to conveniently avoid real numbers. Then, thanks to Gaussian Sampling [12,34], we define an average-case distribution for LIP and establish a worst-case to average-case reduction within an isomorphism class (Sect. 3). This addresses the concerns raised by Blanks and Miller [6], and formalizes their heuristic countermeasure.
- The above cryptographic foundations are directly inspired by the Zero-Knowledge proof of lattice non-isomorphism of Haviv and Regev [16]. We further extend on their techniques by proposing a Zero-Knowledge proof of knowledge (ZKPoK) of a lattice isomorphism (Sect. 4). This directly implies an identification scheme based on sLIP.
- We propose a KEM scheme (Sect. 5) and a hash-then-sign signature scheme (Sect. 6), both based on ΔLIP. Perhaps surprisingly, and unlike the original scheme of McEliece for codes, we circumvent the additional assumption that decoding a certain class of random lattices would be hard. This is done via a lossyness argument [36] for the KEM, and a dual argument for the signature scheme.
- We review the state of the art for solving LIP (Sect. 7). In particular we note that all known algorithms go through lattice reduction, and we quantify the required approximation factor.
- We discuss natural instantiations for each scheme (Sect. 8) from any remarkable lattice. This section handles the construction of the auxiliary lattice appearing in ΔLIP for the lossyness arguments to get through.

1.2 Potential Advantages

The KEM. To instantiate our KEM, consider a lattice L (w.l.o.g. of volume 1) such that:

- the minimal distance is within a factor f from Minkowski's bound: $\lambda_1(L) \geqslant \Omega(\sqrt{n}/f)$,
- there exists an efficient algorithm that can decode errors in L up to radius ρ within a factor f' from Minkowski's bound: $\rho \geqslant \Omega(\sqrt{n}/f')$.[2]

[2] Note that uniqueness of decoding implies $f' \geqslant 2f$.

– the dual minimal distance is within a factor f^* from Minkowski's bound: $\lambda_1(L^*) \geqslant \Omega(\sqrt{n}/f^*)$.

Then, our instantiated KEM appears to resist lattice attack down to an approximation factor $O(\max(f, f^*) \cdot f^* \cdot f')$. More specifically, it's security is based on ΔLIP for two lattices whose primals and duals are all within a factor $O(\max(f, f^*) \cdot f^* \cdot f')$ from Minkowski's bound.

The trivial lattice $\mathbb{Z}^n$ gives all three factors f, f', f^* of the order $\Theta(\sqrt{n})$. The Barnes-Wall [28] lattice improves all three factors down to $\Theta(\sqrt[4]{n})$.

The endgame would be to instantiate with lattices for which all three factors would be very small. In particular, one would naturally turn to recent work on decoding the Chor-Rivest lattices [8,11,20,24] and the Barnes-Sloane lattices [31] giving $f = \text{polylog}(n)$ and $f' = \text{polylog}(n)$, but unfortunately their dual are not that good: $f^* \geqslant \Theta(\sqrt{n})$. Indeed, all these constructions are integer lattices $L \subset \mathbb{Z}^n$ with single exponential volume $\det(L) = c^n$: their dual L^* have a Minkowski's bound of $\Theta(\sqrt{n}/\det(L)^{1/n}) = \Theta(\sqrt{n})$, but contain all of $\mathbb{Z}^n \subset L^*$, including vectors of norm 1.

Note nevertheless that there is no geometric impossibility to the existence of the desired remarkably decodable lattice: random lattices have $f = O(1)$ and $f^* = O(1)$; so decoding is possible down to $f' = O(1)$ but the best known algorithm is conjectured to take exponential time.

The Signature Scheme. The same principle also applies to our signature scheme, but this time with respect to Gaussian sampling rather than decoding: lattices with tight sampling (and large dual minimal distance) would lead to a scheme resisting attacks down to very small approximation factors. Alas, even ignoring the constraint on the dual lattice, we do not know of any lattice much better than $\mathbb{Z}^n$ for efficient gaussian sampling. Yet, instantiated with $\mathbb{Z}^n$ our scheme still has an interesting feature: not having to deal with any Gram-Schmidt or Cholesky matrices over the reals. This may be a worthy practical advantage over current hash-then-sign signature schemes [12].

The Identification Scheme. Because sLIP seems super-exponentially hard in the dimension for well chosen lattices (large kissing number), it might be secure to instantiate our ZKPoK with a rather small lattice dimension, maybe down to about a hundred. Yet, this is more a theoretical curiosity than a practical advantage—the protocol still needs soundness amplification, and each round requires exchanging $\tilde{O}(n^2)$ bits.

1.3 Open Questions

A KEM with polylog*-Approximation Factor Security.* Is there any family of lattices that can be efficiently decoded within a polylog factor from Minkowski's bound such as [8,11,20,24,31], but whose dual would also have an equally large minimal distance?

Tight Gaussian Sampling for Signatures. Is there any family of lattices L (of volume 1) in which one can efficiently sample Gaussian with small parameter $\sigma < o(\sqrt{n})$, if not $\sigma = \text{polylog}(n)$ (with exponential smoothing $\sigma > \eta_{2^{-n}}(L)$)? And if so, do they and their dual have a large minimal distance? Note that quantumly, this question is related to the previous one via the reduction of Regev [39]: decoding in the primal for a large radius gives Gaussian sampling in the dual for a small width. But a classical algorithm would be much preferable.

Concrete Instantiation with Simple Lattices. Instantiated with $\mathbb{Z}^n$, our signature scheme has the advantage of not requiring any Gram-Schmidt or Cholesky decomposition, contrary to existing hash-then-sign signature schemes; and may therefore be of practical interest. It could also be reasonable to instantiate our KEM with the lattice of Barnes and Wall, thanks to the decoder of Micciancio and Nicolesi [28].

Module-LIP. At last, it also seems natural to explore structured variants of LIP, where both the lattice and the isometry should be structured. We note that for any ideal lattice in complex-multiplication number fields, a classical polynomial time algorithm is known [13,23]. Could the module variant be secure? Can our constructions gain a linear factor on key sizes from this variant? And are there remarkably decodable lattices that are also ideals in certain number fields? The repeated-difference lattices (a.k.a. Craig's lattices [9]) are indeed ideal lattices in cyclotomic number field with large minimal distances, but a polynomial decoding algorithm for them remains to be discovered.

2 Preliminaries

2.1 Notation

Vectors $\boldsymbol{x}$ are denoted in bold and should be interpreted as column vectors. For a matrix B with columns $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ we denote its Gram-Schmidt orthogonalisation by B^* with columns $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$, and we denote the matrix norm by $\|B\| := \max_i \|\boldsymbol{b}_i\|_2$. We denote $\mathbb{T}_q$ the discretized torus $\mathbb{T}_q := (\frac{1}{q}\mathbb{Z})/\mathbb{Z}$ and identify it with its set of reduced representatives $\{0, \frac{1}{q}, \ldots, \frac{q-1}{q}\}$. The statistical distance between two random variable X and Y will be denoted $\Delta(X, Y)$.

2.2 Lattice Isomorphism and Quadratic Forms

Abstractly, the set of (full-rank, n-dimensional) lattices can be thought as the homogeneous space[3] $\mathcal{GL}_n(\mathbb{R})/\mathcal{GL}_n(\mathbb{Z})$: a lattice $L = \mathcal{L}(B) := B \cdot \mathbb{Z}^n$ is generated by the columns of a basis $B \in \mathcal{GL}_n(\mathbb{R})$, and two basis $B, B' \in \mathcal{GL}_n(\mathbb{R})$ generate the same lattice if and only if there exists a unimodular matrix $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B' = BU$.

[3] This quotient should read as the quotient of a *set* by the action of group, and not a group quotient. Indeed $\mathcal{GL}_n(\mathbb{Z})$ is *not* a normal subgroup of $\mathcal{GL}_n(\mathbb{R})$ for $n > 1$.

Two lattices are *isomorphic* if there exists an orthonormal transformation $O \in \mathcal{O}_n(\mathbb{R})$ sending one to the other. Finding this transformation, if it exists, is known as the Lattice Isomorphism Problem (LIP).

Definition 2.1 (LIP, lattice version). *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 $\mathcal{L}' = O \cdot \mathcal{L}$.*

Algorithmically lattices $\mathcal{L} = \mathcal{L}(B), \mathcal{L}' = \mathcal{L}(B')$ are represented by bases $B, B' \in \mathcal{GL}_n(\mathbb{R})$, and if $\mathcal{L}' = O \cdot \mathcal{L}$, then OB is a basis of $\mathcal{L}'$. If $OB = B'$, then we can easily compute $O := B'B^{-1}$, however in general OB will only be equal to B' up to some unimodular transformation. More specifically when $\mathcal{L} = \mathcal{L}(B)$, and $\mathcal{L}' = \mathcal{L}(B')$ for some lattice bases $B, B' \in \mathcal{GL}_n(\mathbb{R})$ the Lattice Isomorphism Problem asks to find an orthonormal $O \in \mathcal{O}_n(\mathbb{R})$ *and* a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B' = OBU$. The presence of both the orthonormal and the unimodular transformation is what makes LIP a hard problem. In other words, reconstructing (or even testing) equivalence in either quotient $\mathcal{GL}_n(\mathbb{R})/\mathcal{GL}_n(\mathbb{Z})$ or $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})$ is easy, doing so in the double quotient $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})/\mathcal{GL}_n(\mathbb{Z})$ appears to be hard.

The real-valued coordinates of the basis and orthonormal transformation can be inconvenient and inefficient to work with. We can alleviate some of these concerns by moving to the (equivalent) quadratic form setting, where instead of a basis B we focus on the Gram matrix $Q = B^tB$.

Quadratic Forms and Integral Equivalence. The idea of the Quadratic Form point of view on LIP is to consider the quotient in the opposite order than in the lattice point of view: first on the left by $\mathcal{O}_n(\mathbb{R})$ and then only on the right by $\mathcal{GL}_n(\mathbb{Z})$.

We define a *quadratic form* as a positive definite real symmetric matrix. A quadratic form can be thought as a basis modulo rotation; they realize the quotient $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})$. More precisely, consider the surjective Gram map $\gamma : \mathcal{GL}_n(\mathbb{R}) \to \mathcal{S}_n^{>0}(\mathbb{R})$ sending a lattice basis B to the quadratic form $Q = B^tB$. Note that the preimages of $\gamma(B)$ are precisely the OB for $O \in \mathcal{O}_n(\mathbb{R})$.

For a lattice basis B the Gram matrix $Q = B^tB$ naturally gives a quadratic form. Additionally every quadratic form Q induces a unique upper-triangular lattice basis B_Q such that $Q = B_Q^tB_Q$ (Cholesky decomposition). In the quadratic form setting lattice vectors $B\boldsymbol{x} \in \mathbb{R}^n$ are represented by their integral basis coefficients $\boldsymbol{x} \in \mathbb{Z}^n$. The inner product with respect to a quadratic form is naturally given by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_Q := \boldsymbol{x}^tQ\boldsymbol{y}$, and the norm by $\|\boldsymbol{x}\|_Q^2 := \boldsymbol{x}^tQ\boldsymbol{x}$. Note that this perfectly coincides with the geometry between the original lattice vectors. We denote the ball of radius r by $\mathcal{B}_Q(r) := \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_Q \leqslant r\}$. Translating the lattice definition, one get the *first minimum* $\lambda_1(Q)$ defined by

$$\lambda_1(Q) := \min_{\boldsymbol{x} \in \mathbb{Z}^n \setminus \{0\}} \|\boldsymbol{x}\|_Q ,$$

and more generally the i-th minimal distance $\lambda_i(Q)$ defined as the smallest $r > 0$ such that $\{\boldsymbol{x} \in \mathbb{Z}^n \mid \|\boldsymbol{x}\|_Q \leqslant r\}$ spans a space of dimension at least i.

In this realization $\mathcal{S}_n^{>0}(\mathbb{R})$ of the quotient $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})$, the action of $U \in \mathcal{GL}_n(\mathbb{Z})$ is given by $Q \mapsto U^tQU$. We may now rephrase LIP for two lattice bases B and B'. Note that if $B' = OBU$, then for $Q := B^tB$ we have:

$$Q' := (B')^tB' = U^tB^tO^tOBU = U^tB^tBU = U^tQU,$$

and we call Q and Q' equivalent if such a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ exists, and denote the equivalence class by $[Q]$, moving the real-valued orthonormal transform $O \in \mathcal{O}_n(\mathbb{R})$ out of the picture. Additionally many remarkable lattices attain a rational-valued Gram matrix Q, removing the need for real-valued or approximate arithmetic. Later in this work we will restrict ourselves to integer-valued quadratic forms.

Weaker Equivalence (Genus). The study of integral equivalence of quadratic forms is classically approached via weaker notions, namely, equivalence over larger rings [9, Chapter 15, Sec 4]. In particular, we shall consider the rational equivalence class $[Q]_{\mathbb{Q}}$ of all U^tQU for $U \in \mathcal{GL}_n(\mathbb{Q})$, as well as the p-adic integer equivalence class $[Q]_{\mathbb{Z}_p}$ of all U^tQU for $U \in \mathcal{GL}_n(\mathbb{Z}_p)$. These equivalences are coarser than integral equivalence: $[Q] = [Q'] \Rightarrow [Q]_{\mathbb{Q}} = [Q']_{\mathbb{Q}}$ and $[Q]_{\mathbb{Z}_p} = [Q']_{\mathbb{Z}_p}$. These data $([Q]_{\mathbb{Q}}, ([Q]_{\mathbb{Z}_p})_p)$ about a quadratic form are called the *genus* of the quadratic form.

One could also consider equivalence over the reals $\mathbb{R}$, or over the p-adic rationals $\mathbb{Q}_p$. By a local-global principle (Minkowski-Hasse Theorem [42, Thm. 9, pp. 44]) these data are redundant with the rational class $[Q]_{\mathbb{Q}}$.

The Lattice Isomorphism Problem, Quadratic Form Formulation. The Lattice Isomorphism Problem can now be restated. We start by properly defining the worst-case problems, in both a search and distinguishing variant.

Definition 2.2 (wc-sLIP^Q)**.** *For a quadratic form $Q \in \mathcal{S}_n^{>0}$ the problem* wc-sLIP^Q *is, given any quadratic form $Q' \in [Q]$, to find a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q' = U^tQU$.*

Note that the problem is equivalent to the original LIP problem as we can still extract an orthonormal transformation by computing $O = B'(BU)^{-1}$. Moreover, the automorphism group $\mathrm{Aut}(Q) := \{V \in \mathcal{GL}_n(\mathbb{Z}) : V^tQV = Q\}$ is finite, and for any solution $U \in \mathcal{GL}_n(\mathbb{Z})$ to wc-sLIP^Q such that $Q' = U^tQU$, the full set of solutions is given by $\{VU : V \in \mathrm{Aut}(Q)\}$.

We also consider a *distinguishing* variant of LIP, denoted wc-ΔLIP. It is not to be confused with the *decisional* version of LIP (which we will refer to as dLIP).[4]

[4] In dLIP^{Q_0} one is given an arbitrary Q' and must decide whether Q' belongs to $[Q_0]$. The distinguishing version is potentially easier in that Q' is promised to belong to either $[Q_0]$ or $[Q_1]$ for some known fixed $[Q_1]$.

Definition 2.3 (wc-$\Delta\mathrm{LIP}^{Q_0,Q_1}$)**.** *For two quadratic forms $Q_0, Q_1 \in \mathcal{S}_n^{>0}$ the problem* wc-$\Delta\mathrm{LIP}^{Q_0,Q_1}$ *is, given any quadratic form $Q' \in [Q_b]$ where $b \in \{0,1\}$ is a uniform random bit, to find b.*

Because (part of) the genus is efficiently computable (see Sect. 7), we will make sure that $[Q_0]_R = [Q_1]_R$ for all relevant ring extensions $R \in \{\mathbb{Q}, \mathbb{R}, \mathbb{Q}_p, \mathbb{Z}_p\}$.

Hardness Statements. When we discuss the hardness of LIP problems, we will implicitly assume that we are not talking of a single quadratic form Q (or of a single pair (Q_0, Q_1) for ΔLIP), but of a family $(Q_n)_n$ (or a family of pairs $(Q_{0,n}, Q_{1,n})_n$ for ΔLIP) where n ranges over an infinite set of positive integer.

2.3 Discrete Gaussians and Sampling

Discrete Gaussian sampling has been fundamental to the development of lattice based cryptography, by allowing to return short or nearby lattice vectors without leaking information about the secret key [12]. We rephrase the relevant definitions and propositions in the quadratic form language.

Distribution. For any quadratic form $Q \in \mathcal{S}_n^{>0}$ we define the Gaussian function on $\mathbb{R}^n$ with parameter $s > 0$ and center $\boldsymbol{c} \in \mathbb{R}^n$ by

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

The discrete Gaussian distribution is obtained by restricting the continuous Gaussian distribution to a discrete lattice. In the quadratic form setting the discrete lattice will always be $\mathbb{Z}^n$, but with the geometry induced by the quadratic form. For any quadratic form $Q \in \mathcal{S}_n^{>0}$ we define the discrete Gaussian distribution $\mathcal{D}_{Q,s,\boldsymbol{c}}$ with center $\boldsymbol{c} \in \mathbb{R}^n$ and parameter $s > 0$ by

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

If the center $\boldsymbol{c}$ is not denoted we have $\boldsymbol{c} = \boldsymbol{0}$. An important property of the discrete gaussian distribution is the smoothing parameter, i.e. how much gaussian noise $s > 0$ is needed to 'smooth out' the discrete structure.

Definition 2.4 (Smoothing Parameter). *For a quadratic form $Q \in \mathcal{S}_n^{>0}$ and $\epsilon > 0$ we define the smoothing parameter $\eta_\epsilon(Q)$ as the minimal $s > 0$ such that $\rho_{Q^{-1},1/s}(\mathbb{Z}^n) \leqslant 1 + \epsilon$.*

The smoothing parameter is a central quantity for gaussians over lattice, for example it permits to control the variations of $\rho_{Q,s,\boldsymbol{c}}(\mathbb{Z}^n)$ is over all centers $\boldsymbol{c}$.

Lemma 2.5 ([29])**.** *For any quadratic form $Q \in \mathcal{S}_n^{>0}$, $\epsilon > 0$, center $\boldsymbol{c} \in \mathbb{R}^n$ and parameter $s > \eta_\epsilon(Q)$ we have:*

$$(1-\epsilon)\frac{s^n}{\sqrt{\det(Q)}} \leqslant \rho_{Q,s,\boldsymbol{c}}(\mathbb{Z}^n) \leqslant (1+\epsilon)\frac{s^n}{\sqrt{\det(Q)}}.$$

Note that the smoothing parameter $\eta_\epsilon(Q)$ is an invariant property of the similarity class $[Q]$, and so we might also denote $\eta_\epsilon([Q])$ for a similarity class. While computing or even approximating the exact smoothing parameter is hard, we can obtain sufficient bounds via the dual form.

Lemma 2.6 (Smoothing bound [29]**).** *For any quadratic form $Q \in \mathcal{S}_n^{>0}$ we have $\eta_{2^{-n}}(Q) \leqslant \sqrt{n}/\lambda_1(Q^{-1})$ and $\eta_\epsilon(Q) \leqslant \|B_Q^*\| \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$ for $\epsilon > 0$.*

Above the smoothing parameter the discrete gaussian distribution is in some sense 'well behaved' and we have the following tailbound that one would expect from a Gaussian distribution.

Lemma 2.7 (Tailbound [30, **Lemma 4.4]).** *For any quadratic form $Q \in \mathcal{S}_n^{>0}$, $\epsilon \in (0,1)$, center $\boldsymbol{c} \in \mathbb{R}^n$ and parameter $s \geqslant \eta_\epsilon(Q)$, we have*

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}_{Q,s,\boldsymbol{c}}}[\|\boldsymbol{x}-\boldsymbol{c}\|_Q > s\sqrt{n}] \leqslant \frac{1+\epsilon}{1-\epsilon}\cdot 2^{-n}.$$

A constant factor above the smoothing parameter we can furthermore lower bound the min-entropy of the distribution.

Lemma 2.8 (Min-entropy [35]**).** *For any quadratic form $Q \in \mathcal{S}_n^{>0}$, positive $\epsilon > 0$, center $\boldsymbol{c} \in \mathbb{R}^n$, parameter $s \geqslant 2\eta_\epsilon(Q)$, and for every $\boldsymbol{x} \in \mathbb{Z}^n$, we have*

$$\Pr_{X\sim\mathcal{D}_{Q,s,\boldsymbol{c}}}[X = \boldsymbol{x}] \leqslant \frac{1+\epsilon}{1-\epsilon}\cdot 2^{-n}.$$

Gaussian Sampling. While the discrete Gaussian distribution already is an important theoretical tool, for many practical purposes we want to actually sample (close to) the distribution in an efficient manner. In their breakthrough work Gentry et al. [12] showed that Klein's [19] randomized Babai's nearest plane algorithm does exactly that. Given a lattice basis one can sample statistically close to the discrete Gaussian distribution with parameters depending on the shortness of the (Gram-Schmidt) basis; a better reduced basis allows for a lower Gaussian width s. To simplify later proofs we use an exact sampling algorithm by Brakerski et al. [7].

Lemma 2.9 (Discrete Sampling [7, **Lemma 2.3]).** *There is a polynomial-time algorithm* **DiscreteSample**$(Q, s, \boldsymbol{c})$ *that given a quadratic form $Q \in \mathcal{S}_n^{>0}$, center $\boldsymbol{c} \in \mathbb{R}^n$, and a parameter $s \geqslant \|B_Q^*\| \cdot \sqrt{\ln(2n+4)/\pi}$, returns a sample distributed as $\mathcal{D}_{Q,s,\boldsymbol{c}}$.*

2.4 Randomness Extractors

A randomness extractor allows, using a publicly known random seed, to convert a non-uniform randomness source X with high min-entropy $H_\infty(X) := -\log_2(\max_x \Pr[X = x])$ to a near-uniform random variable [3,15].[5]

Definition 2.10 (Extractor). *An efficient function $\mathcal{E} : \mathcal{X} \times \{0,1\}^z \to \{0,1\}^v$ is an (m, ϵ)-extractor, if, for all random variable X distributed over $\mathcal{X}$ and $H_\infty(X) \geqslant m$, it holds that*

$$\Delta\big((Z, \mathcal{E}(X, Z)), (Z, V)\big) \leqslant \epsilon$$

where the seed $Z \leftarrow \mathcal{U}(\{0,1\}^z)$ and $V \leftarrow \mathcal{U}(\{0,1\}^v)$ are drawn uniformly at random, and independently of X.

When instantiating our scheme, we will rely on the existence of an (m, ϵ)-extractor with parameters $m = \Theta(v)$ and $\epsilon = 2^{-\Theta(m)}$.

3 LIP and Self-reducibility

In this section we lay the foundation for using the Lattice Isomorphism Problem in cryptography. We present an average-case distribution for any quadratic form equivalence class, show how to sample from it, and conclude with a worst-case to average-case reduction. Note that the worst-case to average-case reduction is realized *within* an equivalence class.

3.1 An Average-Case Distribution

First we define our average-case distribution within an equivalence class $[Q]$, which can be seen as an extension of the techniques used by Haviv and Regev [17] to show that LIP lies in SZK. While in their work they use a discrete Gaussian sampler [12] to sample a generating set of the lattice, we extend this by a linear algebra step that returns a canonically distributed lattice basis—or in our case a quadratic form.

A posteriori, this algorithm appears very similar to the heuristic approach of [6], but the use of Gaussian sampling formally guarantees that the output distribution solely depends on the lattice and not on the specific input basis—or in our case, depends only on the class of the input quadratic form.

First we consider the linear algebra step, that given a quadratic form and short linearly independent vectors, returns a well reduced equivalent form.

Lemma 3.1 (Adapted from [27, Lemma 7.1]). *There is a polynomial time algorithm $(R, U) \leftarrow \mathbf{Extract}(Q, Y)$ that on input a quadratic form Q, and linearly independent vectors $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \in \mathbb{Z}^{n \times n}$, outputs a transformation $U \in \mathcal{GL}_n(\mathbb{Z})$ and a quadratic form $R = U^t Q U$ equivalent to Q such that $\|B_R^*\| \leqslant \max_i \|\boldsymbol{y}_i\|_Q$.*

[5] For our application, we do not need to relax the source to only have average min-entropy, and therefore work with the simpler worst-case version.

Proof. First let $U \in \mathcal{GL}_n(\mathbb{Z})$ be the unique transformation such that $T = U^{-1}Y$ is the canonical upper-diagonal Hermite Normal Form of Y. Let $R = U^tQU$ and note that R is equivalent to Q. Denote the column vectors of U by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$. Because T is upper triangular and in Hermite Normal Form we have $\boldsymbol{y}_i = \sum_{j=1}^{i} T_{j,i}\boldsymbol{u}_j$, where $T_{j,j} \geqslant 1$. In particular we have that $\mathrm{span}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k) = \mathrm{span}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$. Let $\boldsymbol{y}_i^*$ and $\boldsymbol{u}_i^*$ be the i-th Gram-Schmidt vector of Y and U respectively w.r.t. Q. Note that $\boldsymbol{y}_i^* = T_{i,i} \cdot \boldsymbol{u}_i^*$, and thus $\|\boldsymbol{u}_i^*\|_Q = \|\boldsymbol{y}_i^*\|_Q / T_{i,i} \leqslant \|\boldsymbol{y}_i^*\|_Q \leqslant \|\boldsymbol{y}_i\|_Q$. We conclude by $\|B_R^*\| = \max_i \|\boldsymbol{u}_i^*\|_Q \leqslant \max_i \|\boldsymbol{y}_i\|_Q$. □

For our final distribution to be well defined we need that the extracted quadratic form only depends on the geometry of the input vectors, and not on the particular representative Q.

Lemma 3.2 (Independence of representative). *Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \in \mathbb{Z}^n$ be linearly independent. If $(R, U) \leftarrow$ **Extract**(Q, Y), and for some unimodular $V \in \mathcal{GL}_n(\mathbb{Z})$ we have $(R', U') \leftarrow$ **Extract**$(V^tQV, V^{-1}Y)$, then $R' = R$, and $U' = V^{-1} \cdot U$.*

Proof. From the canonicity of the Hermite Normal Form we immediately obtain that $(U')^{-1}V^{-1}Y = T = U^{-1}Y$, and thus $U' = V^{-1} \cdot U$. It follows that $R' = (V^{-1} \cdot U)^t V^t Q V (V^{-1} \cdot U) = U^tQU = R$. □

Now we can formally define our average-case distribution for a parameter $s > 0$.

Definition 3.3. *Given a quadratic form equivalence class $[Q] \subset \mathcal{S}_n^{>0}$ we define the Gaussian form distribution $\mathcal{D}_s([Q])$ over $[Q]$ with parameter $s > 0$ algorithmically as follows:*

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

By Lemma 3.2 the output is independent of the chosen representative and thus the distribution is well-defined.

Given the algorithmic definition of $\mathcal{D}_s([Q])$, an actual efficient sampling algorithm follows with only a few adaptations. Firstly, we need to efficiently sample from $\mathcal{D}_{Q,s}$ which puts some constraints on the parameter s depending on the reducedness of the representative Q. Secondly the probability that n sampled vectors are linearly independent can be quite small, instead we sample vectors one by one and only add them to our set Y if they are independent. Still we require the additional constraint $s \geqslant \lambda_n(Q)$ to show that this succeeds with a polynomial amount of samples.

Lemma 3.4. *For any quadratic form $Q \in \mathcal{S}_n^{>0}(\mathbb{Z})$, and parameter*

$$s \geqslant \max\{\lambda_n(Q), \|B_Q^*\| \cdot \sqrt{\ln(2n+4)/\pi}\},$$

Algorithm 1: Sampling from $\mathcal{D}_s([Q])$.

Data: A quadratic form $Q \in \mathcal{S}_n^{>0}(\mathbb{Z}^n)$, and a parameter $s \geqslant \max\{\lambda_n(Q), \|B_Q^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}$.
Result: Sample $R = U^tQU$ from $\mathcal{D}_s([Q])$, with a transformation $U \in \mathcal{GL}_n(\mathbb{Z})$.
$Y \leftarrow \emptyset$;
while $|Y| < n$ **do**
  $\boldsymbol{x} \leftarrow \mathcal{D}_{Q,s}$; // Using Lemma 2.9
  if $\boldsymbol{x} \notin \mathrm{span}(Y)$ **then**
    Append $\boldsymbol{x}$ to Y;
  end
end
$(R, U) \leftarrow$ **Extract**(Q, Y);

Algorithm 1 runs in expected polynomial time and returns (R, U) where R is a sample from $\mathcal{D}_s([Q])$, and a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $R = U^tQU$. Furthermore, the isomorphism U is uniform over the set of isomorphisms from Q to R.

Proof. By Lemmas 2.9 and 3.1 every step in Algorithm 1 runs in polynomial time. What remains is to show that the number of iterations is polynomially bounded. Let the random variable K be the number of samples before we find n independent ones. If $|Y| < n$, then because $s \geqslant \lambda_n(Q)$ we have by [17, Lemma 5.1] that every newly sampled vector $\boldsymbol{x} \leftarrow \mathcal{D}_{Q,s}$ is not in the span of Y with constant probability at least $C := 1 - (1 + e^{-\pi})^{-1} > 0$. So K is bounded from above by a negative binomial distribution for n successes with success probability C, which implies that $\mathbb{E}[K] \leqslant \frac{n}{C}$, and in particular that $\Pr[K > n^2] \leqslant e^{-\Omega(n^2)}$. When the while loop succeeds the set Y is distributed as n vectors sampled from $\mathcal{D}_{Q,s}$ under the linear independence condition, following exactly Definition 3.3.

Suppose that the algorithm runs and finishes with a final spanning set Y, and returning $(R, U) \leftarrow$ **Extract**(Q, Y). For any automorphism $V \in \mathrm{Aut}(Q)$, i.e. such that $V^tQV = Q$, it would have been just as likely that the final spanning set equalled VY, because the samples from $\mathcal{D}_{Q,s}$ only depend on the norm of the vectors w.r.t. Q. Then by Lemma 3.2 we have:

$$\textbf{Extract}(Q, VY) = \textbf{Extract}((V^{-1})^tQV^{-1}, VY) = (R, VU),$$

and thus the algorithm would have returned VU with the same probability as U, which makes the returned transformation uniform over the set of isomorphisms $\{VU : V \in \mathrm{Aut}(Q)\}$ from Q to R. □

For (exponentially) large parameters s we can always efficiently sample from the average-case distribution by first LLL-reducing the representative.

Lemma 3.5. *Given any quadratic form $Q \in \mathcal{S}_n^{>0}(\mathbb{Z})$ we can sample from $\mathcal{D}_s([Q])$ in polynomial time for $s \geqslant 2^{\Theta(n)} \cdot \lambda_n([Q])$.*

Proof. Run the LLL algorithm on Q to obtain a representative $Q' \in [Q]$ for which $\|B^*_{Q'}\| \leqslant 2^{\Theta(n)} \cdot \lambda_n([Q])$. Then apply Lemma 3.4. □

Lemma 3.6. *For any quadratic form $Q \in \mathcal{S}_n^{>0}$, parameter $\epsilon \in (0,1)$, and $s \geqslant \max\{\lambda_n(Q), \eta_\epsilon(Q)\}$, we have*

$$\Pr_{Q' \sim \mathcal{D}_s([Q])}[\|B^*_{Q'}\| > s\sqrt{n}] \leqslant \frac{1+\epsilon}{1-\epsilon} \cdot 25n \cdot 2^{-n}.$$

Proof. Given linearly independent vectors $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \in \mathbb{Z}^n$ the extractor returns a quadratic form Q' such that $\|B^*_{Q'}\| \leqslant \max_i \|\boldsymbol{y}_i\|_Q$ and thus we can just focus on the norms $\|\boldsymbol{y}_i\|_Q$. Let the random variable K be the number of samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K \leftarrow \mathcal{D}_{Q,s}$ before we find n independent ones. By Lemma 2.7 we have $\|\boldsymbol{x}_i\| > s\sqrt{n}$ with probability at most $(1+\epsilon)/(1-\epsilon) \cdot 2^{-n}$. By the proof of Lemma 3.4 we have $\mathbb{E}[K] \leqslant \frac{n}{C} \leqslant 25n$, and by a union bound we conclude:

$$\begin{aligned}\Pr\left[\max_i \|\boldsymbol{y}_i\|_Q > s\sqrt{n}\right] &= \sum_{k=n}^{\infty} \Pr[K=k] \Pr\left[\max_{1 \leqslant i \leqslant k} \|\boldsymbol{x}_i\|_Q > s\sqrt{n}\right] \\ &\leqslant \underbrace{\sum_{k=n}^{\infty} \Pr[K=k] \cdot k}_{\mathbb{E}[K]} \cdot \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n} \leqslant \frac{n}{C} \cdot \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}.\end{aligned}$$

□

3.2 Average Case LIP

The above definition of a distribution over a class which is efficiently sampleable from any representative of that class leads us to a natural average-case version of both version of LIP. It is parametrized by a width parameter $s > 0$.

Definition 3.7 (ac-sLIP_s^Q)**.** *For a quadratic form $Q \in \mathcal{S}_n^{>0}$ and $s > 0$ the problem* ac-sLIP_s^Q *is, given a quadratic form sampled as $Q' \leftarrow \mathcal{D}_s([Q])$, to find a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q' = U^t Q U$.*

Definition 3.8 ($\text{ac-}\Delta\text{LIP}_s^{Q_0,Q_1}$)**.** *For two quadratic forms $Q_0, Q_1 \in \mathcal{S}_n^{>0}$ and $s > 0$ the problem* $\text{ac-}\Delta\text{LIP}_s^{Q_0,Q_1}$ *is, given a quadratic form sampled as $Q' \leftarrow \mathcal{D}_s([Q_b])$ where $b \in \{0,1\}$ is a uniform random bit, to find b.*

Trivially the average-case variants can be reduced to their respective worst-case variants. In the following section we show that the reverse is also true.

3.3 A Worst-Case to Average-Case Reduction

In general lattice problems become easier when given a short basis; and harder when given a long basis. Similarly one would expect that ac-sLIP_s^Q and $\text{ac-}\Delta\text{LIP}_s^{Q_0,Q_1}$ become harder when the parameter $s > 0$ increases. In fact when s is large enough the average-case problem becomes at least as hard as any worst-case instance; making the average-case and worst-case problems equivalent.

Lemma 3.9 (ac-sLIP$_s^Q \geqslant$ wc-sLIP^Q for large s). *Given an oracle that solves* ac-sLIP$_s^Q$ *for some $s \geqslant 2^{\Theta(n)} \cdot \lambda_n(Q)$ in time T_0 with probability $\epsilon > 0$, we can solve* wc-sLIP^Q *with probability at least ϵ in time $T = T_0 + poly(n, \log s)$.*

Proof. Given any $Q' \in [Q]$, apply Lemma 3.5 to sample $Q'' \leftarrow \mathcal{D}_s([Q])$ for some $s \geqslant 2^{O(n)} \cdot \lambda_n([Q])$, together with a U'' such that $Q'' = U''^t Q' U''$. Note that $\mathcal{D}_s([Q]) = \mathcal{D}_s([Q'])$; we can therefore apply our ac-sLIP$_s^Q$-oracle to Q'' and obtain $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q'' = U^t Q U$. Now for $U' := UU''^{-1} \in \mathcal{GL}_n(\mathbb{Z})$ we have:

$$U'^t Q U' = (U''^{-1})^t U^t Q U U''^{-1} = (U''^{-1})^t Q'' U''^{-1} = Q'.$$

So given an ac-sLIP$_s^Q$-oracle we can solve wc-sLIP^Q. □

To allow for more efficient schemes we would like to decrease the parameter $s > 0$ in the worst-case to average-case reduction. We can do so at the cost of stronger lattice reduction than LLL.

Lemma 3.10. *Given an oracle that solves* ac-sLIP$_s^Q$ *for some $s \geqslant \lambda_n(Q)$ in time T_0 with probability $\epsilon > 0$, we can solve* wc-sLIP^Q *with probability at least $\frac{1}{2}$ in time*

$$T = \frac{1}{\epsilon}(T_0 + \text{poly}(n, \log s)) + C\left(n, \frac{s}{\lambda_n(Q) \cdot \sqrt{\ln(2n+4)/\pi}}\right),$$

where $C(n, f)$ is the cost of solving the Shortest Independent Vector Problem (SIVP, [39]) within an approximation factor of f.

Proof. The f-approx-SIVP oracle returns n linearly independent vectors of norm at most $f \cdot \lambda_n(Q)$, and thus using Lemma 3.1 we can construct an equivalent form $Q' \in [Q]$ with $\|B^*_{Q'}\| \leqslant f \cdot \lambda_n(Q)$. For $f := s/(\lambda_n(Q) \cdot \sqrt{\ln(2n+4)/\pi})$ we obtain that $s \geqslant \|B^*_{Q'}\| \cdot \sqrt{\ln(2n+4)/\pi}$, and thus we can sample efficiently from $\mathcal{D}_s([Q])$. The rest of the proofs follows similar to that of Lemma 3.9. Additionally the reduction succeeds with some probability $\epsilon > 0$, so we need to repeat it $\frac{1}{\epsilon}$ times to obtain a success probability of at least $\frac{1}{2}$. Note that each additional sample can be computed in polynomial time from the same representative Q'. □

Remark 3.11. Note that the overhead is entirely additive, in particular it does not suffer from the $\frac{1}{\epsilon}$ amplification. So, while the reduction is not polynomial time, concretely, one can afford huge overheads; for example an overhead of 2^{100} would barely affect a underlying hardness of 2^{128} as $2^{128} - 2^{100} = 2^{127.999\ldots}$. This situation is quite different from the usual innefficient reductions found in the literature, where the overhead is multiplicative.

In Lemma 3.10, the SIVP oracle can be instantiated by a variant of the BKZ algorithm [40]. With a sub-linear blocksize of $\beta := n/\log(n)$ we could decrease s to a quasi-polynomial factor $\exp(\log^2(n)) \cdot \lambda_n(Q)$, with only a sub-exponential additive cost to the reduction. For security based on exponential hardness (e.g. $T_0/\epsilon = \exp(\Omega(n))$) this would still be meaningful, while maintaining a poly-logarithmic bitlength for the integer entries of the manipulated matrices.

Going down to polynomial factors $s = \mathrm{poly}(n) \cdot \lambda_n(Q)$ (and hence single logarithmic integer bitlength) would require a linear blocksize $\beta := \Theta(n)$, and an exponential cost 2^{cn}. For small constants $c > 0$ such that cn is smaller than the security parameter the reduction would still be meaningful. However for provable algorithms this constant c is rather large, and the gap between provable [1] and heuristic results [4] is significant. As we want to keep our reduction non-heuristic in this initial work, we will leave this regime for further research.

Using a similar strategy, one can also establish a worst-case to average-case reduction for ΔLIP. Note that, because it is a distinguishing problem, the advantage amplification now requires $O(1/\alpha^2)$ calls to the average-case oracle.

Lemma 3.12 (ac-ΔLIP$_s^{Q_0,Q_1} \geqslant$ wc-ΔLIP^{Q_0,Q_1} for large s). *Given an oracle that solves* ac-ΔLIP$_s^{Q_0,Q_1}$ *for some* $s \geqslant 2^{\Theta(n)} \cdot \max\{\lambda_n(Q_0), \lambda_n(Q_1)\}$ *in time* T_0 *with advantage* $\alpha > 0$*, we can solve* wc-ΔLIP^{Q_0,Q_1} *with advantage* α *in time* $T + poly(n, \log s)$.

Lemma 3.13. *Given an oracle that solves* ac-ΔLIP$_s^{Q_0,Q_1}$ *in time* T_0 *for some* $s \geqslant \max\{\lambda_n(Q_0), \lambda_n(Q_1)\}$ *with advantage* $\alpha > 0$*, we can solve* wc-ΔLIP^{Q_0,Q_1} *with advantage at least* $\frac{1}{4}$ *in time*

$$T = \frac{1}{\alpha^2}(T_0 + \mathrm{poly}(n, \log s)) + C\left(n, \frac{s}{\max\{\lambda_n(Q_0), \lambda_n(Q_1)\} \cdot \sqrt{\ln(2n+4)/\pi}}\right),$$

where $C(n, f)$ *is the cost of solving the Shortest Independent Vector Problem (SIVP, [39]) within an approximation factor of* f.

4 Zero Knowledge Proof of Knowledge

At high level, the protocol of Haviv and Regev [17], as well as ours, is very similar to protocols for other types of isomorphisms, in particular protocols for graph ismorphism [14] and for code isomorphism [5].

A notable difference however, is that both these protocols [5,14] relied on the action of a finite group (permutations), allowing to show zero-knowledgness by uniformity of the distribution over an orbit. In our case, the group acting $\mathcal{GL}_n(\mathbb{Z})$ is not finite, and not even compact, admitting no such uniform distribution. It is perhaps surprising to see that uniformity is in fact not required.

4.1 The Σ-Protocol

Efficiency and Completeness. For efficiency of Σ we have to check that Algorithm 1 runs in polynomial time, and indeed by Lemma 3.4 this is the case because

$$s \geqslant \max\left\{\lambda_n([Q_0]), \|B^*_{Q_0}\| \cdot \sqrt{\ln(2n+4)/\pi}\right\}.$$

Zero Knowledge Proof of Knowledge Σ

Consider two equivalent public quadratic forms $Q_0, Q_1 \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and a secret unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q_1 = U^t Q_0 U$. Given the public parameter

$$s \geqslant \max\left\{\lambda_n([Q_0]), \max\left\{\|B^*_{Q_0}\|, \|B^*_{Q_1}\|\right\} \cdot \sqrt{\ln(2n+4)/\pi}\right\},$$

we define the following protocol Σ that gives a zero-knowledge proof of knowledge of an isomorphism between Q_0 and Q_1:

Prover		Verifier
Sample $Q' \leftarrow \mathcal{D}_s([Q_0])$ by Alg. 1, together with V s.t. $Q' = V^t Q_0 V$		
	$\xrightarrow{Q'}$	Sample $c \leftarrow \mathcal{U}(\{0,1\})$
Compute $W = U^{-c} \cdot V$	$\xleftarrow{c}$	
	$\xrightarrow{W}$	Check if $W \in \mathcal{GL}_n(\mathbb{Z})$, and $Q' = W^t Q_c W$.

For the verification we have that $W \in \mathcal{GL}_n(\mathbb{Z})$ if and only if W is integral and $\det(W) = \pm 1$, both of which are easy to check in polynomial time.

For the completeness of Σ note that when the prover executes the protocol honestly we have $W := U^{-c} \cdot V \in \mathcal{GL}_n(\mathbb{Z})$ because U and V are both unimodular by definition. Additionally we have

$$Q' = V^t Q_0 V = \underbrace{(V^t (U^{-c})^t)}_{W^t} \underbrace{((U^c)^t Q_0 U^c)}_{Q_c} \underbrace{(U^{-c} V)}_{W} = W^t Q_c W,$$

and thus the verifier accepts.

Special Soundness. Suppose we have two accepting conversations $(Q', 0, W_0)$ and $(Q', 1, W_1)$ of Σ where the first message is identical. The acceptance implies that $W_0, W_1 \in \mathcal{GL}_n(\mathbb{Z})$ and $W_0^t Q_0 W_0 = Q' = W_1^t Q_1 W_1$, and thus $U' := W_0 W_1^{-1} \in \mathcal{GL}_n(\mathbb{Z})$ gives an isomorphism from Q_0 to Q_1 as

$$U'^t Q_0 U' = (W_1^{-1})^t (W_0^t Q_0 W_0) W_1^{-1} = (W_1^{-1})^t (W_1^t Q_1 W_1) W_1^{-1} = Q_1.$$

We conclude that Σ has the special soundness property.

Special Honest-Verifier Zero-Knowledge. We create a simulator that given the public input Q_0, Q_1 outputs an accepting conversation with the same probability distribution as between a honest prover and verifier. Note that the first message Q' is always distributed as $\mathcal{D}_s([Q_0])$, the challenge c as $\mathcal{U}(\{0,1\})$, and V is uniform over the set of isomorphisms from Q_0 to Q' by Lemma 3.4. Because U is an isomorphism from Q_0 to Q_1 we have, given the challenge c, that $W = U^{-c} \cdot V$ is uniform over the set of isomorphisms from Q_c to Q'.

To simulate this we first sample the uniformly random challenge $c \leftarrow \mathcal{U}(\{0,1\})$. If $c = 0$ we can proceed the same as in Σ itself, e.g. sample

$Q' \leftarrow \mathcal{D}_s([Q_0])$ using Algorithm 1, together with a V such that $Q' = V^t Q_0 V$, and set $W := V$. The final conversation $(Q', 0, W)$ is accepting and follows by construction the same distribution as during an honest execution conditioned on challenge $c = 0$.

If $c = 1$ we use the fact that $[Q_0] = [Q_1]$, and that we can use Algorithm 1 with representative Q_1 as input instead of Q_0. So again we obtain $Q' \leftarrow \mathcal{D}_s([Q_1]) = \mathcal{D}_s([Q_0])$ following the same distribution, but now together with a unimodular $W \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q' = W^t Q_1 W$. The conversation $(Q', 1, W)$ is accepting by construction, and Q' follows the same distribution $\mathcal{D}_s([Q_0])$. Additionally by Lemma 3.4 the transformation W is indeed uniform over the set of isomorphisms from Q_1 to Q'.

We conclude that Σ has the special honest-verifier zero-knowledge property.

4.2 Identification Scheme

The Zero Knowledge Proof of Knowledge in the previous section is worst-case in the sense that given any two equivalent forms $Q_0, Q_1 \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and a secret isomorphism $U \in \mathcal{GL}_n(\mathbb{Z})$ from Q_0 to Q_1 we can show knowledge of such an isomorphism. However to turn this Σ-protocol into an Identification Scheme (see e.g. [10]) we need to define a distribution of $U \in \mathcal{GL}_n(\mathbb{Z})$ (or alternatively of Q_1 w.r.t Q_0). Finding an isomorphism between Q_0 and Q_1 is at most as hard as solving either ac-sLIP$_s^{Q_0}$ or ac-sLIP$_s^{Q_1}$ for parameter s as in Σ. Therefore a natural choice is to have Q_1 distributed according to $\mathcal{D}_{s'}([Q_0])$ for some parameter $s' \geqslant \max\{\lambda_n([Q_0]), \|B_{Q_0}^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}$, which we can efficiently sample from using Algorithm 1. The security of our identification scheme is then solely based on the hardness of ac-sLIP$_{s'}^{Q_0}$.

5 Key Encapsulation Mechanism

In this section we construct a Key Encapsulation Mechanism (KEM) with a security proof based on the hardness of ΔLIP. In short we will need a quadratic form S along with an efficient decoder up to some radius $\rho < \lambda_1(S)/2$. The public key will consist of a long equivalent form $P := U^t S U \leftarrow \mathcal{D}_s([S])$, while the unimodular transformation U will be the secret key. Knowledge of the transformation U allows to decode w.r.t. P via S; without any loss in decoding performance. The key will be a random error $\boldsymbol{e}$ of norm $\|\boldsymbol{e}\|_P \leqslant \rho$, and it can be encapsulated as the syndrome $\overline{\boldsymbol{e}} := \boldsymbol{e} \bmod \mathbb{Z}^n \in [0,1)^n$. The receiver with knowledge of the secret transformation U can recover $\boldsymbol{e}$ by decoding via S. The correctness follows from the fact that the decoding is unique due to $\rho < \lambda_1(S)/2$.

For the security we assume that it is (computationally) hard to differentiate between $P \leftarrow \mathcal{D}_s([S])$ and some random sample $R \leftarrow \mathcal{D}_s([Q])$ from a special class $[Q]$, a class corresponding to a lattice admitting a dense sublattice. This assumption allows us to replace P by R, which completely breaks the uniqueness of the decoding. That is, the syndrome $\overline{\boldsymbol{e}}$ has many (say $\exp \Omega(\lambda)$) nearby points w.r.t. R, and retrieving the exact original point becomes statistically hard.

Key Encapsulation Scheme

Let $\rho < \lambda_1(S)/2$ and let $S \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a quadratic form with an efficient decoder **Decode** with decoding radius ρ. Let $\mathcal{E} : \frac{1}{q}\mathbb{Z}^n \times \{0,1\}^z \to \{0,1\}^\ell$ be a $(\ell, \mathrm{negl}(n))$-extractor for some $\ell = \Theta(n)$. Given the public parameters

$$s \geqslant \max\{\lambda_n(S), \|B_S^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}, \text{ and}$$
$$q := \left\lceil \frac{s \cdot n}{\rho} \cdot \sqrt{\ln(2n+4)/\pi} \right\rceil,$$

we define the KEM $\mathcal{K} := (\mathbf{Gen}, \mathbf{Encaps}, \mathbf{Decaps})$ as follows:

- $(pk, sk) \leftarrow \mathbf{Gen}(1^n)$: on input 1^n do:
  1. Sample $P \leftarrow \mathcal{D}_s([S])$ using Alg. 1, together with U such that $P = U^t S U$.
  2. Output $(pk, sk) = (P, U)$.
- $(c, k) \leftarrow \mathbf{Encaps}(pk)$: on input 1^n and a public key $P = pk$ do:
  1. Sample $\boldsymbol{e} \leftarrow \frac{1}{q}\mathcal{D}_{P, q\rho/\sqrt{n}} \in \frac{1}{q}\mathbb{Z}^n$ using Lemma 2.9.
  2. Compute $\boldsymbol{c} \leftarrow \boldsymbol{e} \bmod \mathbb{Z}^n$ s.t. $\boldsymbol{c} \in \mathbb{T}_q^n = \{0, \frac{1}{q}, \ldots, \frac{q-1}{q}\}^n$.
  3. Sample a random extractor seed $Z \leftarrow \{0,1\}^z$.
  4. Compute $k \leftarrow \mathcal{E}(\boldsymbol{e}, Z)$.
  5. Output (c, k) where $c := (\boldsymbol{c}, Z)$.
- $k \leftarrow \mathbf{Decaps}(sk, c)$: on input $c = (\boldsymbol{c}, Z)$ and a secret key $U := sk$ do:
  1. Compute $\boldsymbol{y} \leftarrow \mathbf{Decode}(S, U\boldsymbol{c})$ s.t. $\|\boldsymbol{y} - U\boldsymbol{c}\|_S \leqslant \rho$, output $\perp$ on failure.
  2. Compute $k \leftarrow \mathcal{E}(\boldsymbol{c} - U^{-1}\boldsymbol{y}, Z)$.
  3. Output k.

Efficiency and Correctness. We consider the efficiency and correctness of the KEM $\mathcal{K} := (\mathbf{Gen}, \mathbf{Encaps}, \mathbf{Decaps})$ instantiated with quadratic form $S \in \mathcal{S}_n^{>0}(\mathbb{Z})$ and public parameter

$$s \geqslant \max\{\lambda_n(S), \|B_S^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}.$$

By the above constraint on s, Algorithm 1 will run in polynomial-time by Lemma 3.4. Furthermore by Lemma 3.6 we have with overwhelming prob ability that

$$q\rho/\sqrt{n} \geqslant s\sqrt{n} \cdot \sqrt{\ln(2n+4)/\pi} \geqslant \|B_P^*\| \cdot \sqrt{\ln(2n+4)/\pi},$$

and thus we can efficiently sample from $\mathcal{D}_{P, q\rho/\sqrt{n}}$ by Lemma 2.9.

For correctness note that in the key encapsulation algorithm the sampled error $\boldsymbol{e}$ has norm at most $\|\boldsymbol{e}\|_P \leqslant \rho$ except with negligible probability by Lemma 2.7, and we denote the encapsulated key by $k := \mathcal{E}(\boldsymbol{e}, Z)$, where Z denotes the randomness extractor's seed. Because $\rho < \lambda_1(S)/2$ the vector $\boldsymbol{x} := \boldsymbol{c} - \boldsymbol{e} \in \mathbb{Z}^n$ is the unique closest vector to $\boldsymbol{c}$ with respect to P, which makes $U\boldsymbol{x}$ the unique closest vector to $U\boldsymbol{c}$ with respect to $S = (U^{-1})^t P U^{-1}$. In the decapsulation the decoder computes the unique vector $\boldsymbol{y}$ at distance at most ρ from $U\boldsymbol{c}$, which implies that $\boldsymbol{y} = U\boldsymbol{x}$. So indeed the output $k' := \mathcal{E}(\boldsymbol{c} - U^{-1}\boldsymbol{y}, Z) = \mathcal{E}(\boldsymbol{c} - \boldsymbol{x}, Z) = \mathcal{E}(\boldsymbol{e}, Z) = k$ equals the encapsulated key with overwhelming probability.

CPA Security. To show that our KEM is CPA-secure we fall back to a lossy trapdoor argument a la [36]. Under the hardness of decisional LIP we can replace our unique ρ-decodable quadratic form by one that is far from uniquely decodable. For the latter it is enough to have a dense sublattice.

Lemma 5.1. *Let $Q \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a quadratic form with a rank r sublattice $D\mathbb{Z}^r \subset \mathbb{Z}^n$. For positive $\epsilon > 0$, center $\boldsymbol{c} \in \mathbb{R}^n$, parameter $s := \rho/\sqrt{n} \geqslant 2\eta_\epsilon([D^tQD])$, and for every $\boldsymbol{x} \in \mathbb{Z}^n$ we have*

$$\Pr_{X \sim \mathcal{D}_{Q,s,\boldsymbol{c}}}[X = \boldsymbol{x}] \leqslant \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-r}.$$

Proof. Let $\boldsymbol{y} := \boldsymbol{x} - \boldsymbol{c} \in \mathbb{R}^n$, and decompose $\boldsymbol{y} =: \boldsymbol{y}_D + \boldsymbol{y}_{D^\perp}$ where $\boldsymbol{y}_D \in \mathrm{span}(D\mathbb{Z}^r)$, and $\boldsymbol{y}_{D^\perp}$ is orthogonal to $\boldsymbol{y}_D$ w.r.t Q. Then we have

$$\begin{aligned}\Pr_{X \sim \mathcal{D}_{Q,s,\boldsymbol{c}}}[X = \boldsymbol{x}] &= \frac{\rho_{Q,s,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{Q,s,\boldsymbol{c}}(\mathbb{Z}^n)} = \frac{\rho_{Q,s}(\boldsymbol{y})}{\rho_{Q,s}(\boldsymbol{y}+\mathbb{Z}^n)} \leqslant \frac{\rho_{Q,s}(\boldsymbol{y})}{\rho_{Q,s}(\boldsymbol{y}+D\mathbb{Z}^r)} \\ &= \frac{\rho_{Q,s}(\boldsymbol{y}_{D^\perp}) \cdot \rho_{Q,s}(\boldsymbol{y}_D)}{\rho_{Q,s}(\boldsymbol{y}_{D^\perp}) \cdot \rho_{Q,s}(\boldsymbol{y}_D + D\mathbb{Z}^r)} = \frac{\rho_{Q,s}(\boldsymbol{y}_D)}{\rho_{Q,s}(\boldsymbol{y}_D + D\mathbb{Z}^r)}.\end{aligned}$$

Note that we can write $\boldsymbol{y}_D = D\boldsymbol{z}$ for some $\boldsymbol{z} \in \mathbb{R}^r$, then the above equals $\Pr_{X \sim \mathcal{D}_{D^tQD,s,\boldsymbol{z}}}[X = 0]$, which by Lemma 2.8 is bounded by $\frac{1+\epsilon}{1-\epsilon} \cdot 2^{-r}$. □

Theorem 5.2. *We consider the KEM $\mathcal{K} := (\mathbf{Gen}, \mathbf{Encaps}, \mathbf{Decaps})$ instantiated with quadratic form $S \in \mathcal{S}_n^{>0}(\mathbb{Z})$, decoding radius ρ, and public key parameter $s > 0$. Let $Q \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a quadratic form with a dense rank $r = \Theta(n)$ sublattice $D\mathbb{Z}^r \subset \mathbb{Z}^n$, in particular such that $\eta_{\frac{1}{2}}(D^tQD) \leqslant \rho/(2\sqrt{n})$. Then $\mathcal{K}$ is CPA-secure if* ac-$\Delta\mathrm{LIP}_s^{S,Q}$ *is hard.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary. We present two games $\mathbf{Game}_1$ and $\mathbf{Game}_2$, where $\mathbf{Game}_1$ is the regular CPA-security game with the original scheme, and $\mathbf{Game}_2$ is almost identical but with the only change that the public key is drawn from $\mathcal{D}_s([Q])$ instead of $\mathcal{D}_s([S])$. By the hardness of ac-$\Delta\mathrm{LIP}_s^{S,Q}$ the two games are computationally indistinguishable, and due to the dense sublattice we can conclude that winning $\mathbf{Game}_2$ with a non-negligible advantage is statistically impossible.

Let the key-size $\ell = \Theta(n)$ be such that $\ell \leqslant r - \log_2(3)$. The original KEM CPA game $\mathbf{Game}_1$ is as follows [18]:

- $\mathbf{Gen}(1^n)$ is run to obtain a public key $pk = P$. Then $\mathbf{Encaps}(pk)$ is run to generate (c, k) with $k \in \{0,1\}^\ell$.
- A uniform bit $b \in \{0,1\}$ is chosen. If $b = 0$, set $\hat{k} := k$, if $b = 1$, choose a uniform $\hat{k} \in \{0,1\}^\ell$.
- Given $(pk, c = (\boldsymbol{c}, Z), \hat{k})$ the adversary $\mathcal{A}$ wins the experiment if b is guessed correctly.

The only difference between $\mathbf{Game}_1$ and $\mathbf{Game}_2$ is that in $\mathbf{Game}_2$ we sample the public key P from $\mathcal{D}_s([Q])$ instead of $\mathcal{D}_s([S])$. Note that $\mathbf{Game}_1$ and $\mathbf{Game}_2$

both only use public information and thus by the hardness of ac-$\Delta\mathrm{LIP}_s^{S,Q}$ the two are computationally indistinguishable by $\mathcal{A}$.

Now we take a look at $\mathbf{Game}_2$. Consider the output $(c = (\boldsymbol{c}, Z), k) \leftarrow \mathbf{Encaps}(pk)$ where $pk := Q' \in [Q]$. For any fixed $\boldsymbol{c}$ we have by construction that $k := \mathcal{E}(\boldsymbol{e}, Z)$, where $\boldsymbol{e} \leftarrow \frac{1}{q}\mathcal{D}_{Q',q\rho/\sqrt{n}}$ under the condition that $\boldsymbol{e} = \boldsymbol{c} \bmod \mathbb{Z}^n$. Equivalently we could say that $\boldsymbol{e} \leftarrow \boldsymbol{c} - \mathcal{D}_{Q',\rho/\sqrt{n},\boldsymbol{c}}$, then by Lemma 5.1 we know that $\boldsymbol{e}$ has a min-entropy of at least $r - \log_2(3) \geqslant l$, and thus $k := \mathcal{E}(\boldsymbol{e}, Z) \in \{0,1\}^\ell$ is negligibly close to uniform independent of c. So in $\mathbf{Game}_2$ we have that $\hat{k}$ is negligibly close to uniform, independent of c and the choice of $b \in \{0,1\}$, making it impossible for $\mathcal{A}$ to guess b with non-negligible advantage. □

6 Signature Scheme

Similar to the Key Encapsulation Mechanism we propose in this section a *hash-then-sign* signature scheme based on ΔLIP. The main requirement is a quadratic form S along with an efficient discrete Gaussian sampling algorithm of smallish width $\rho/\sqrt{n} \geqslant \eta_{2^{-\Theta(n)}}(S)$.

Again the public key will consist of some lesser reduced form $P := U^t S U \leftarrow \mathcal{D}_s([S])$ equivalent to S, where the unimodular transformation U is the secret key. To sign a message we use a full domain hash to obtain a uniform coset $\boldsymbol{t} + \mathbb{Z}^n$, the signature then consists of a nearby vector $\boldsymbol{\sigma} \leftarrow \mathcal{D}_{P,\rho/\sqrt{n},\boldsymbol{t}}$ w.r.t. the form P. The nearby vector is obtained via S by the secret transformation U.

The security assumption is similar, but in some way dual to that of the KEM. Again assume that it is computationally hard to differentiate between P and some special class of forms $[Q]$; however in this case Q must admit a sparse projection (equivalently, their dual should contain a dense lattice). The sparsity implies that a uniformly random target $\boldsymbol{t}$ does not have a nearby vector with overwhelming probability, making the signage vacuously hard.

Correctness. For correctness we mainly have to check that the returned signature $\boldsymbol{\sigma} \in \mathbb{Z}^n$ is indeed close to $\boldsymbol{t} := \mathcal{H}(m)$ w.r.t P. Because $P = U^t S U$ we have:

$$\|\boldsymbol{\sigma} - \boldsymbol{t}\|_P = \|U(\boldsymbol{\sigma} - \boldsymbol{t})\|_S = \|\boldsymbol{\sigma}' - U\boldsymbol{t}\|_S ,$$

and by Lemma 2.7 we have with overwhelming probability that $\|\boldsymbol{\sigma} - \boldsymbol{t}\|_P = \|\boldsymbol{\sigma}' - U\boldsymbol{t}\|_S \leqslant \rho/\sqrt{n} \cdot \sqrt{n} = \rho$, concluding the correctness.

Security. For the security proof we first consider a class of quadratic forms for which the signage is vacuously hard, e.g. for a random target $\boldsymbol{t} \in \mathbb{R}^n/\mathbb{Z}^n$ there exists no nearby vector.

Lemma 6.1. *Let $Q \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a quadratic form with a dense rank k sublattice $D\mathbb{Z}^k \subset \mathbb{Z}^n$, in particular such that $\rho/\sqrt{k} \leqslant 1/(\sqrt{8\pi e} \cdot \det(D^t Q D)^{1/2k})$. Then for the dual form Q^{-1} we have*

$$\Pr_{\boldsymbol{t} \sim \mathcal{U}([0,1]^n)}[|(\boldsymbol{t} + \mathcal{B}^n_{Q^{-1},\rho}) \cap \mathbb{Z}^n| \geqslant 1] \leqslant 2^{-k}.$$

Signature Scheme

Let $S \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a quadratic form together with a sampling algorithm **DiscreteSample** that allows to sample statistically close to $\mathcal{D}_{P,\rho/\sqrt{n}}(\boldsymbol{t}+\mathbb{Z}^n)$ for some parameter $\rho/\sqrt{n} \geqslant \eta_{2^{-\Theta(n)}}([S])$ and any target $\boldsymbol{t} \in \mathbb{T}_q^n$. Let $\mathcal{H} : \mathcal{M} \to \mathbb{T}_q^n$ be a full domain hash function (modeled as a random oracle). Given the public parameters

$$s \geqslant \max\{\lambda_n(S), \|B_S^*\| \cdot \sqrt{\ln(2n+4)/\pi}\}, \text{ and}$$
$$q := \left\lceil \frac{s \cdot n}{\rho} \cdot \sqrt{\ln(2n+4)/\pi} \right\rceil,$$

we define the signature scheme $\mathcal{S} := (\mathbf{Gen}, \mathbf{Sign}, \mathbf{Verify})$ as follows:

- $(pk, sk) \leftarrow \mathbf{Gen}(1^n)$: on input 1^n do:
  1. Sample $P \leftarrow \mathcal{D}_s([S])$ using Alg. 1, together with U s.t. $P = U^t S U$.
  2. Output $(pk, sk) = (P, U) \in \mathcal{S}_n^{>0}(\mathbb{Z}) \times \mathcal{GL}_n(\mathbb{Z})$.
- $\boldsymbol{\sigma} \leftarrow \mathbf{Sign}(sk, m)$: on input a message m and a secret key $U := sk$ do:
  1. Compute $\boldsymbol{t} \leftarrow \mathcal{H}(m)$.
  2. Sample $\boldsymbol{\sigma}' \leftarrow \mathcal{D}_{S,\rho/\sqrt{n},U\boldsymbol{t}}$ using **DiscreteSample**.
  3. Compute $\boldsymbol{\sigma} \leftarrow U^{-1}\boldsymbol{\sigma}'$.
  4. Output $\boldsymbol{\sigma} \in \mathbb{Z}^n$.
- $b := \mathbf{Verify}(pk, m, \boldsymbol{\sigma})$: on input a public key $P = pk$, a message m and a signature $\boldsymbol{\sigma}$ do:
  1. Compute $\boldsymbol{t} \leftarrow \mathcal{H}(m)$.
  2. If $\boldsymbol{\sigma} \in \mathbb{Z}^n$, and $\|\boldsymbol{t} - \boldsymbol{\sigma}\|_P \leqslant \rho$, output $b = 1$.
  3. Otherwise, output $b = 0$.

Proof. Let $V := \mathrm{span}(D) \subset \mathbb{R}^n$ such that the orthogonal projection w.r.t. Q^{-1} of $\mathbb{Z}^n$ onto V defines a projected lattice $C\mathbb{Z}^k := \pi_{Q^{-1},V}(\mathbb{Z}^n)$ of rank k, with $\det(C^t Q^{-1} C) \geqslant 1/\det(D^t Q D)$. Because a projection is non-increasing in length we have

$$\Pr_{\boldsymbol{t} \sim \mathcal{U}(\mathbb{R}^n/\mathbb{Z}^n)}[|(\boldsymbol{t} + \mathcal{B}^n_{Q^{-1},\rho}) \cap \mathbb{Z}^n| \geqslant 1] \leqslant \Pr_{\boldsymbol{t} \sim \mathcal{U}(\mathbb{R}^k/\mathbb{Z}^k)}[|(\boldsymbol{t} + \mathcal{B}^k_{C^t Q^{-1} C,\rho}) \cap \mathbb{Z}^n| \geqslant 1] = (*).$$

Then using Markov's inequality we can bound the above by

$$(*) \leqslant \mathbb{E}_{\boldsymbol{t} \sim \mathcal{U}(\mathbb{R}^k/\mathbb{Z}^k)}[|(\boldsymbol{t} + \mathcal{B}^k_{C^t Q^{-1} C,\rho}) \cap \mathbb{Z}^n|] = \frac{\mathrm{Vol}_{C^t Q^{-1} C}(\mathcal{B}^k_{C^t Q^{-1} C,\rho})}{\mathrm{Vol}_{C^t Q^{-1} C}(\mathbb{R}^k/\mathbb{Z}^k)}$$
$$\leqslant \frac{(2\pi e/k)^{k/2} \cdot \rho^k}{\sqrt{\det(C^t Q^{-1} C)}} \leqslant 2^{-k}.$$

□

Theorem 6.2. *We consider the signature scheme* $\mathcal{S} := (\mathbf{Gen}, \mathbf{Sign}, \mathbf{Verify})$ *instantiated with quadratic form* $S \in \mathcal{S}_n^{>0}(\mathbb{Z})$*, sampling parameter* ρ*, and public key parameter* $s > 0$*. Let* $Q \in \mathcal{S}_n^{>0}(\mathbb{Z})$ *be a quadratic form with a dense rank* $k = \Theta(n)$ *sublattice* $D\mathbb{Z}^k \subset \mathbb{Z}^n$*, in particular such that* $2\rho/\sqrt{k} \leqslant (\sqrt{8\pi e} \cdot \det(D^t Q D^t)^{1/k})^{-1}$*. Then* $\mathcal{S}$ *is EUF-CMA secure if* ac-$\Delta\mathrm{LIP}_s^{S,Q^{-1}}$ *is hard.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary. We present three games $\mathbf{Game}_1, \mathbf{Game}_2, \mathbf{Game}_3$ where $\mathbf{Game}_1$ is the regular EUF-CMA game with the original scheme, $\mathbf{Game}_2$ reprograms the random oracle to generate valid signatures without knowledge of the secret key, and $\mathbf{Game}_3$ samples the public key from $[Q^{-1}]$ instead of $[S]$. By a standard smoothness argument the adversary's view of $\mathbf{Game}_1$ and $\mathbf{Game}_2$ is statistically indistinguishable, and $\mathbf{Game}_2$ and $\mathbf{Game}_3$ are indistinguishable by the hardness of ac-$\Delta\mathrm{LIP}_s^{S,Q^{-1}}$. Then we conclude by Lemma 6.1 that the problem of forging a signature in $\mathbf{Game}_3$ is statistically hard. The original EUF-CMA game $\mathbf{Game}_1$ is as follows [18]:

- $\mathbf{Gen}(1^n)$ is run to obtain keys $(pk = P, sk = U)$.
- Adversary $\mathcal{A}$ is given $pk = P$ and access to an oracle $\mathbf{Sign}(sk, \cdot)$. The adversary then outputs $(m, \boldsymbol{\sigma})$ where m was not queried before to the oracle.
- $\mathcal{A}$ succeeds if and only if $\mathbf{Verify}(pk, m, \boldsymbol{\sigma}) = 1$.

To show that our signature scheme $\mathcal{S}$ is EUF-CMA secure we have to show that $\mathbf{Game}_1$ succeeds only with negligible probability. We assume that the adversary queries the oracle on $l = \mathrm{poly}(n)$ distinct[6] message $m_1, \ldots, m_l$. In $\mathbf{Game}_1$ the secret key is used to obtain a valid signature $(m_i, \boldsymbol{\sigma}_i)$ where $\boldsymbol{\sigma}_i \leftarrow \mathcal{D}_{P,\rho/\sqrt{n},\mathcal{H}(m_i)}$. In $\mathbf{Game}_2$ instead we first sample a random error $\boldsymbol{e}_i \leftarrow \frac{1}{q} \cdot \mathcal{D}_{P,q\rho/\sqrt{n}}$. By Lemma 3.6 we have $q\rho/\sqrt{n} \geqslant \|B_P^*\| \cdot \sqrt{\ln(2n+4)/\pi}$ with overwhelming probability, and thus by Lemma 2.9 we can do the sampling without using the secret key. Then we reprogram the random oracle such that $\mathcal{H}(m_i) := \boldsymbol{t}_i = \boldsymbol{e} \bmod \mathbb{Z}^n \in \mathbb{T}_q$, and return the signature pair $(m_i, \boldsymbol{\sigma}_i := \boldsymbol{t}_i - \boldsymbol{e}_i)$. Note that the probability that $\boldsymbol{t}_i$ equals any target $\boldsymbol{t} \in \mathbb{T}_q^n$ is proportional to $\rho_{P,\rho/\sqrt{n},\boldsymbol{t}}(\mathbb{Z}^n)$. So $\boldsymbol{t}_i$ is close to uniform by Lemma 2.5 because $\rho/\sqrt{n} \geqslant \eta_{2^{-\Theta(n)}}([S]) = \eta_{2^{-\Theta(n)}}([P])$, and thus the random oracle is still simulated correctly. Additionally the conditional probability of $\boldsymbol{\sigma}_i$ conditioned on $\boldsymbol{t}_i$ is exactly the same as in $\mathbf{Game}_1$, so we can conclude that $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are statistically indistinguishable from the adversary's point of view.

The only difference between $\mathbf{Game}_2$ and $\mathbf{Game}_3$ is that in $\mathbf{Game}_3$ we sample the public key P from $\mathcal{D}_s([Q^{-1}])$ instead of $\mathcal{D}_s([S])$. Note that $\mathbf{Game}_2$ and $\mathbf{Game}_3$ both only use public information and thus by the hardness of ac-$\Delta\mathrm{LIP}_s^{S,Q^{-1}}$ the two are computationally indistinguishable.

To conclude note that for any message m we obtain a random target $\boldsymbol{t} := \mathcal{H}(m) \in \mathbb{T}_q^n$. Let $\boldsymbol{e}'$ be uniform over the Babai nearest plane region defined by P, then $\|\boldsymbol{e}'\|_P \leqslant \frac{\sqrt{n}}{2}\|B_P^*\|$, and $\boldsymbol{t}' := \boldsymbol{t} + \frac{1}{q}\boldsymbol{e}'$ is uniform over $\mathbb{R}^n/\mathbb{Z}^n$. By Lemma 6.1 the uniformly random target $\boldsymbol{t}'$ lies at distance at least 2ρ from $\mathbb{Z}^n$ w.r.t. P with overwhelming probability. So for $\boldsymbol{t}$ we have with overwhelming probability that:

$$\begin{aligned}\mathrm{dist}_P(\boldsymbol{t}, \mathbb{Z}^n) &\geqslant \mathrm{dist}_P(\boldsymbol{t}', \mathbb{Z}^n) - \left\|\frac{1}{q}\boldsymbol{e}'\right\|_P \geqslant 2\rho - \frac{\sqrt{n} \cdot \|B_P^*\|}{2q} \\ &\geqslant 2\rho - \rho/(2\sqrt{\ln(2n+4)/\pi}) > \rho.\end{aligned}$$

Therefore it is statistically impossible for the adversary to return a valid signature for m, and thus to win $\mathbf{Game}_3$. □

[6] This can be enforced by salting messages or by derandomization.

7 Cryptanalysis of LIP

Equivalent quadratic forms $Q, Q' := U^t Q U$ (for some $U \in \mathcal{GL}_n(\mathbb{Z})$) share many common properties, and these invariants can be used to decide that two quadratic forms cannot be equivalent, or can guide the search for an isomorphism.

7.1 Invariants

Arithmetic Invariants. Firstly we have $\det(U) = \pm 1$, and thus for two equivalent quadratic forms we have $\det(Q') = \det(U^t)\det(Q)\det(U) = \det(Q)$. Secondly because U and U^{-1} are both integral, the quantity $\gcd(Q) = \gcd\{Q_{ij} : 1 \leqslant i, j \leqslant n\}$ is also an invariant.

A third and less obvious invariant is the parity of the quadratic form. The notion is standard for unimodular lattices: it is called even if all norms are even, and odd otherwise. More generally, writing $\|\boldsymbol{x}\|_Q = \sum_i Q_{ii} x_i^2 + 2\sum_{i<j} x_j Q_{ij} x_i$ one gets that $\gcd\{\|\boldsymbol{x}\|_Q : x \in \mathbb{Z}^n\} \in \{1, 2\} \cdot \gcd(Q)$. We call this factor $\mathrm{par}(Q) \in \{1, 2\}$ the parity of Q. It is also efficiently computable by noting that $\mathrm{par}(Q) = \gcd(\{Q_{ii} : 1 \leqslant i \leqslant n\} \cup \{2\gcd(Q)\})/\gcd(Q)$.

Further arithmetic invariants are induced by R-equivalence of quadratic forms for extensions $R \supset \mathbb{Z}$. The invariants for $\mathbb{Q}$-equivalence can be decomposed via a local-global principle, namely the Hasse-Minkowski theorem [42, Thm. 9, pp. 44]. Together with the discriminant, these invariants are complete (they entirely determine quadratic forms up to $\mathbb{Q}$-equivalence), and they can be computed efficiently. They consists of the signature, and the Cassel-Hasse invariant at each prime p. The Sylvester signature ($\mathbb{R}$-equivalence) is always $(n, 0)$ in our case as we are only considering positive quadratic forms. The Cassel-Hasse invariant ($\mathbb{Q}_p$-invariance) for a prime p is given for a diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_n)$ by

$$h_p = \prod_{i<j} (d_i, d_j)_p \tag{1}$$

where $(\cdot, \cdot)_p$ denotes the Hilbert Symbol at p. Using LDL^t decomposition (Cholesky decomposition over the rationals), one can efficiently compute Hasse invariant for any positive quadratic form.

Similarly, there are also invariants induced by p-adic equivalence of quadratic forms: $Q' = V^t Q V$ for $V \in \mathcal{GL}_n(\mathbb{Z}_p)$, see [9, Chap. 15, Sec 4.1].

All these arithmetic invariants provide a fingerprint

$$\mathrm{ari}(Q) = (\det(Q), \gcd(Q), \mathrm{par}(Q), [Q]_{\mathbb{Q}}, ([Q]_{\mathbb{Z}_p})_p) \tag{2}$$

and they appear to all be efficiently computable, but are essentially only useful to answer the ΔLIP problem in the negative. When instantiating ΔLIP, we should therefore make sure that these fingerprint matches.

The Hull. In the literature for linear code equivalence a relevant notion is that of the efficiently computable hull $C \cap C^\perp$ of a code $C \subset \mathbb{F}_q^n$. Properties such as the rank of the hull are invariant under equivalence, and a small rank even allows to efficiently find the isometry [41]. For a lattice $\mathcal{L}$ and its dual $\mathcal{L}^*$ we could define the hull as $\mathcal{L} \cap \mathcal{L}^*$. However, for integral lattices (or more generally if the associated quadratic form is integral) we always have $\mathcal{L} \subset \mathcal{L}^*$ and thus the hull $\mathcal{L} \cap \mathcal{L}^* = \mathcal{L}$ does not present us with new information. We could generalize definition to consider $\mathcal{L} \cap (k \cdot \mathcal{L}^*)$ for rational $k \in \mathbb{Q}_{\neq 0}$, and although we do not see a direct threat for our instantiation (in Sect. 8) from this, we do encourage more research into the geometric properties of the resulting lattices.

Geometric Invariant. The defining and most important property of a unimodular transformation $U \in \mathcal{GL}_n(\mathbb{Z})$ is that it gives a bijection $\mathbb{Z}^n \to \mathbb{Z}^n$ by $\boldsymbol{x} \mapsto U\boldsymbol{x}$ (or $\boldsymbol{x} \mapsto U^{-1}\boldsymbol{x}$). With respect to the quadratic forms $Q, Q' := U^t Q U$ this even gives an isometry (from Q' to Q) as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{Q'} = \boldsymbol{x}^t Q' \boldsymbol{y} = \boldsymbol{x}^t U^t Q U \boldsymbol{y} = \langle U\boldsymbol{x}, U\boldsymbol{y} \rangle_Q \text{ for } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

This isometry results in several natural geometric invariants related to the norms and inner products of integral vectors. We have already seen some, namely the first minimum $\lambda_1(Q)$ and the i-th minimum $\lambda_i(Q)$. Further geometric invariants can be defined, such as the kissing number $\kappa(Q) = |\mathrm{Min}(Q)|$ where

$$\mathrm{Min}(Q) := \{\boldsymbol{x} \in \mathbb{Z}^n : \|\boldsymbol{x}\|_Q = \lambda_1(Q)\},$$

and more generally the (formal) Theta-series $\Theta_Q(q) = \sum_{\ell \geqslant 0} N_\ell q^\ell$ associated to Q, where $N_\ell = |\{\boldsymbol{x} \in \mathbb{Z}^n : \|\boldsymbol{x}\|_Q = \ell\}|$.

All these geometric invariant appears to involve finding or even enumerating short vectors; in particular they are plausibly hard to compute.

7.2 Algorithms for Distinguish-LIP and Hardness Conjecture

In Sect. 8, we will use ΔLIP with quadratic forms with different minimal distances $\lambda_1(Q_0) < \lambda_1(Q_1)$. However we will be careful to ensure that their arithmetic invariant match $\mathrm{ari}(Q_0) = \mathrm{ari}(Q_1)$ to not make the problem trivial.

Approximate-SVP Oracle. An f-approx-SVP oracle applied to a form Q finds a short vector of length at most $f \cdot \lambda_1(Q)$. So ΔLIP is no harder than f-approx-SVP for $f = \lambda_1(Q_1)/\lambda_1(Q_0)$ in any of those lattices.

Unusual-SVP via Lattice Reduction. However even when the gap between $\lambda_1(Q_0)$ and $\lambda_1(Q_1)$ is small, the minimal vectors may individually still be unusually short, which make them significantly easier to find than in a random lattice. This is usually formalized via the f-unique-SVP problem, but many instances of interest do not have such a gap between λ_1 and λ_2; in fact $\mathbb{Z}^n$, Barnes-Wall and Barnes-Sloane lattices all have $\lambda_1 = \lambda_2 = \cdots = \lambda_n$. But practical and heuristic

studies have showed that uniqueness is not that relevant to lattice attacks [2]. We therefore introduce yet another lattice problem, called *unusual-SVP* to discuss such instances. A formal complexity reduction between unusual-SVP and unique-SVP matching or approaching the heuristic state of the art appears to be a valuable research objective, but is beyond the scope of the present article.

We define f-unusual-SVP: find a minimal vector under the promise that $\lambda_1(Q) \leqslant \mathrm{gh}(Q)/f$, where the Gaussian Heuristic $\mathrm{gh}(Q)$ is a heuristic estimate for $\lambda_1(Q)$ given by:

$$\mathrm{gh}(Q) := \det(Q)^{1/2n} \cdot \frac{1}{\sqrt{\pi}} \cdot \Gamma(1 + n/2)^{1/n} \approx \det(Q)^{1/2n} \cdot \sqrt{\frac{n}{2\pi e}}.$$

State of the art lattice reduction techniques find these unusually short vector more easily than longer vectors with length around $\mathrm{gh}(Q)$, and (heuristically) the hardness is directly driven by the ratio $f = \mathrm{gh}(Q)/\lambda_1(Q)$ [2]. Given a form $Q' \in [Q_0] \cup [Q_1]$ we parametrize the lattice reduction algorithm to find a unusual short vector with length $\min\{\lambda_1(Q_0), \lambda_1(Q_1)\}$, then depending on success we learn that either $Q' \in [Q_0]$ or $Q' \in [Q_1]$.

An Approach of Szydlo. Additionally there is one heuristic algorithm in the literature [46] for ΔLIP, that applies to lattices obtained by mild sparsification of the orthogonal lattice $\mathbb{Z}^n$. This algorithm proceeds by sampling vectors of length $O(\sqrt{n})$ and then decides via a statistical test: the Theta-series appears sufficiently different at such low lengths to distinguish the two lattices. Remarkably, the parameters for state of the art lattice reduction algorithms parametrized to solve $O(\sqrt{n})$-approx-SVP for (mild sparsifications of) $\mathbb{Z}^n$, match those to solve $\mathrm{gh}(\mathbb{Z}^n)/\lambda_1(\mathbb{Z}^n) = O(\sqrt{n})$-unusual-SVP; instead of finding approximate vectors we immediately find the shortest vectors. Again we see that the ratio $\mathrm{gh}(Q)/\lambda_1(Q)$ is what seems to matter.

Conclusion. To conclude, let us also note that any of the above attack can also be run over the dual. To state a hardness conjecture capturing these attacks we define the primal-dual gap to the Gaussian Heuristic as:

$$\mathrm{gap}(Q) = \max\left\{\frac{\mathrm{gh}(Q)}{\lambda_1(Q)}, \frac{\mathrm{gh}(Q^{-1})}{\lambda_1(Q^{-1})}\right\}.$$

Note that this quantity might be slightly lower than 1 (but no lower than 1/2 by Minkowski bound): there might exist excellent lattice packings beating the Gaussian Heuristic. We will be assuming[7] $\mathrm{gap}(Q_i) \geqslant 1$, which implies that $\lambda_1(Q_i)/\lambda_1(Q_{1-i}) \leqslant \mathrm{gap}(Q_i)$, therefore also capturing the first approach.

In all the attacks above, one first searches for vector no larger than $f \cdot \lambda_1(Q_i)$ w.r.t. Q_i for $f = \mathrm{gap}(Q_i)$, hence the following conjecture.

[7] That is, we cowardly shy away from making hardness conjecture on such exceptionally dense lattice packings. Such a regime has never been considered in practical cryptanalysis and would deserve specific attention. We suspect that SVP in such lattices to be even harder than in random lattices.

Conjecture 7.1 (Hardness of ΔLIP *(Strong)).* For any class of quadratic forms $[Q_0], [Q_1]$ of dimension n, with $\mathrm{ari}([Q_0]) = \mathrm{ari}([Q_1])$, $1 \leqslant \mathrm{gap}([Q_i]) \leqslant f$, the best attack against wc-$\Delta\mathrm{LIP}^{Q_0,Q_1}$ requires solving f-approx-SVP in the worst-case from either $[Q_0]$ or $[Q_1]$.

This conjecture is meant to offer a comparison point with existing lattice-based cryptography in terms of the approximating factor. Beyond contradicting this assumption, we also invite cryptanalysis effort toward concrete comparison of f-approx-SVP on those instances to SIS and LWE with the same approximation factor f. If one only wishes to argue exponential security in n of the schemes proposed in this paper, a sufficient conjecture is the following.

Conjecture 7.2 (Hardness of ΔLIP *(Mild)).* For any class of quadratic forms $[Q_0], [Q_1]$ of dimension n, with $\mathrm{ari}([Q_0]) = \mathrm{ari}([Q_1])$, $\mathrm{gap}([Q_i]) \leqslant \mathrm{poly}(n)$, wc-$\Delta\mathrm{LIP}^{Q_0,Q_1}$ is $2^{\Theta(n)}$-hard.

Note that the conjecture above are "best-case" over the choice of the isomorphism class, and worst-case over the representation of the class (however note that we have a worst-case to average-case reduction over that representation). That is, even though we may only want to use ΔLIP for specific choices of isomorphism classes, we gladly invite cryptanalysis effort on ΔLIP on any choice of isomorphism classes.

7.3 Algorithms for Search-LIP and Challenges

While the above invariants allow to semi-decide LIP, the search version requires more effort; though all methods known to us at least require the enumeration of short primal or dual vectors. In the extended version of this work[8] we discuss these methods in more detail.

8 Instantiating ΔLIP Pairs from Remarkable Lattices

To instantiate our schemes, we do not only need a lattice with efficient decoding or sampling; we also need a second lattice with a specific property to instantiate the ΔLIP problem and argue security. This section deals with how the ΔLIP pair is constructed from a single remarkable lattice.

8.1 Key Encapsulation Mechanism

To instantiate our KEM we need two quadratic forms: a form S along with an efficient decoder that can decode up to some distance $\rho < \lambda_1(Q)/2$, and a form Q with a dense rank k sublattice $D \cdot \mathbb{Z}^k \subset \mathbb{Z}^n$ such that $\eta_{\frac{1}{2}}(D^t Q D) \leqslant \rho/(2\sqrt{n})$. For simplicity of notation we move to the lattice point of view.

We assume to have an n-dimensional lattice Λ for which $\mathrm{gap}(\Lambda) \leqslant f = f(n)$, and for which we can decode up to $\rho = \Theta(1/f) \cdot \mathrm{gh}(\Lambda) < \lambda_1(\Lambda)/2$. We

[8] The full version of this work is available at https://eprint.iacr.org/2021/1332.

consider a general construction leading to a $2n$-dimensional primary lattice Λ_S and secondary lattice Λ_Q with gap bounded by $O(f^3)$ and such that Λ_Q has a dense enough sublattice to instantiate our KEM.

Note that due to the bounded gap of Λ we have by Lemma 2.6 that

$$\eta_{\frac{1}{2}}(\Lambda) \leqslant \eta_{2^{-n}}(\Lambda) \leqslant \frac{\sqrt{n}}{\lambda_1(\Lambda^*)} \leqslant \frac{\sqrt{n} \cdot f}{\mathrm{gh}(\Lambda^*)} = \Theta(f \cdot \det(\Lambda)^{1/n}).$$

Now let $g = \Theta(f^2)$ be a positive integer and consider the lattices:

$$\Lambda_S := g \cdot \Lambda \oplus (g+1) \cdot \Lambda, \text{ and } \Lambda_Q := \Lambda \oplus g(g+1)\Lambda,$$

where by construction Λ_Q has a dense sublattice Λ. Note that we can still decode Λ_S up to radius $\rho' := g \cdot \rho = \Theta(g/f) \cdot \mathrm{gh}(\Lambda)$.

Invariants Match. Both lattices have determinant $g^n(g+1)^n \det(\Lambda)^2$. Due to the coprimality of g and $g+1$ we still have $\gcd(\Lambda_S) = \gcd(\Lambda_Q) = \gcd(\Lambda)$, and similarly for the parity. It remains to check rational equivalence and p-adic equivalence for all primes p. Let R denote a quadratic form representing Λ. Up to integral equivalence, we have:

$$S := \begin{pmatrix} g^2 R & 0 \\ 0 & (g+1)^2 R \end{pmatrix} \qquad Q := \begin{pmatrix} R & 0 \\ 0 & g^2(g+1)^2 R \end{pmatrix}.$$

Let I_n be the $n \times n$ identity matrix and consider the transformations:

$$U_1 := \begin{pmatrix} g^{-1} I_n & 0 \\ 0 & g I_n \end{pmatrix} \qquad U_2 := \begin{pmatrix} 0 & (g+1) I_n \\ (g+1)^{-1} I_n & 0 \end{pmatrix}$$

Then $Q = U_1^t S U_1$ over $\mathbb{Q}$: this implies $[S]_{\mathbb{Q}} = [Q]_{\mathbb{Q}}$. For any prime p we have that either $\gcd(g, p) = 1$ or $\gcd(g+1, p) = 1$ (or both). So either g or $(g+1)$ is invertible over the p-adic integers $\mathbb{Z}_p$, and thus either $U_1 \in \mathcal{GL}_d(\mathbb{Z}_p)$ exists and $Q = U_1^t S U_1$ over $\mathbb{Z}_p$ or $U_2 \in \mathcal{GL}_d(\mathbb{Z}_p)$ exists and $Q = U_2^t S U_2$ over $\mathbb{Z}_p$. In either case, we have established $[S]_{\mathbb{Z}_p} = [Q]_{\mathbb{Z}_p}$, which concludes the comparison of arithmetic invariants: $\mathrm{ari}(S) = \mathrm{ari}(Q)$.

Dense Sublattice. We now check the requirements for Theorem 5.2, namely that $\eta_{\frac{1}{2}}(\Lambda) \leqslant \rho'/(2\sqrt{2n})$. Given that $\eta_{\frac{1}{2}}(\Lambda) \leqslant \Theta(f \cdot \mathrm{gh}(\Lambda)/\sqrt{n})$, it is sufficient if

$$\Theta(f \cdot \mathrm{gh}(\Lambda)/\sqrt{n}) \leqslant \rho'/(2\sqrt{2n}) = \Theta(g/f) \cdot \mathrm{gh}(\Lambda)/\sqrt{n},$$

and thus we can conclude that some $g = \Theta(f^2)$ indeed suffices.

Following the conclusions from the cryptanalysis in Sect. 7.2 and more specifically Conjecture 7.1, we take a look at the primal-dual gap for Λ_S and Λ_Q. We have that $\mathrm{gap}(\Lambda_S) = \Theta(\mathrm{gap}(\Lambda)) \leqslant O(f)$, and $\mathrm{gap}(\Lambda_Q) = \Theta(g \cdot \mathrm{gap}(\Lambda)) \leqslant O(f^3)$. Note that following the same computation above but for a primal gap of f, dual gap of f^*, and a decoding gap of $f' \geqslant 2f$ we would have $g = \Theta(f^* \cdot f')$ and obtain a final primal-dual gap of $O(\max(f, f^*) \cdot f^* \cdot f')$.

8.2 Signature Scheme

Our signature scheme can be instantiated with any lattice for which we can sample efficiently at small Gaussian widths, following a similar ΔLIP pair as above. Namely, we assume to have a lattice Λ with $\mathrm{gap}(\Lambda) \leqslant f$ and such that we can sample efficiently with parameter $\rho/\sqrt{n} = \Theta(\eta_{2^{-\Theta(n)}}(\Lambda))$ close to the smoothing bound. Similarly to the KEM we set $\Lambda_S := g \cdot \Lambda \oplus (g+1) \cdot \Lambda$, and $\Lambda_{Q^{-1}} = \Lambda \oplus g(g+1) \cdot \Lambda$ for some integer $g \geqslant 1$. In particular, as in the KEM, we do have $\mathrm{ari}(S) = \mathrm{ari}(Q^{-1})$.

Then for the dual we have $\Lambda_Q = \Lambda^* \oplus \frac{1}{g(g+1)}\Lambda^*$, with $\frac{1}{g(g+1)}\Lambda^*$ as a dense sublattice. The constraint of Theorem 6.2 boils down to the inequality $\Theta(g \cdot f \cdot \det(\Lambda)^{1/n}) \leqslant \Theta(g^2 \det(\Lambda)^{1/n})$, and thus some $g = \Theta(f)$ suffices. The final primal-dual gap of Λ_S and $\Lambda_{Q^{-1}}$ is then bounded by $O(f^2)$.

The simplest lattice for which we have very efficient samplers is of course the integer lattice $\mathbb{Z}^n$, leading to a gap of $O(n)$ via the above construction. Instantiating our scheme with this lattice would lead to an interesting signature scheme where there is no need to compute any Cholesky decomposition, even for signing, and that could be fully implemented with efficient integer arithmetic.

We refer to our last open question (Sect. 1.3) regarding lattices with a tighter Gaussian sampler, in order to obtain a signature scheme with a better underlying approximation factor.

Getting Down to $O(f)$. The general constructions presented turn a good decodable or sampleable lattice Λ with gap f into a primary and secondary lattice with gap $O(f^3)$ and $O(f^2)$ to instantiate our KEM and signature scheme respectively. We suggest here that these losses might be an artifact of the security proof.

Suppose we can generate a random lattice Λ_Q such that $\mathrm{ari}(\Lambda_Q) = \mathrm{ari}(\Lambda)$; without the arithmetic constraint we would have with overwhelming probability that $\mathrm{gap}(\Lambda_Q) = O(1)$ (but even $O(f)$ would suffice). Let's assume that the constraint does not affect this gap. Then similar to the scheme of McEliece, by adding the extra security assumption that it is hard to decode in Λ_Q (or hard to sample for the signature scheme), we could remove the lossyness argument from the security proof and directly instantiate our schemes with the pair (Λ, Λ_Q), leading to a gap of $O(f)$.

Acknowledgments. The authors would like to express their gratitude to Jelle Don, Chris Peikert, Alice Pellet-Mary, Damien Stehlé and Benjamin Wesolowski for relevant discussions and their precious feedback. Special thanks go to Aron van Baarsen for bringing the genus to our attention, and to Thomas Debris-Alazard and Alain Couvreur for bringing the hull to our attention.

The research of L. Ducas was supported by the European Union's H2020 Programme under PROMETHEUS project (grant 780701) and the ERC-StG-ARTICULATE project (no. 947821). W. van Woerden is funded by the ERC-ADG-ALGSTRONGCRYPTO project (no. 740972).

References

1. Aggarwal, D., Dadush, D., Regev, O., Stephens-Davidowitz, N.: Solving the shortest vector problem in 2^n time using discrete Gaussian sampling. In: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, pp. 733–742 (2015)
2. Albrecht, M., Ducas, L.: Lattice attacks on NTRU and LWE: a history of refinements. Cryptology ePrint Archive Report 2021/799 (2021). https://ia.cr/2021/799
3. Barak, B., et al.: Leftover hash lemma, revisited. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 1–20. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_1
4. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 10–24. SIAM (2016)
5. Biasse, J.-F., Micheli, G., Persichetti, E., Santini, P.: LESS is more: code-based signatures without syndromes. In: Nitaj, A., Youssef, A. (eds.) AFRICACRYPT 2020. LNCS, vol. 12174, pp. 45–65. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51938-4_3
6. Blanks, T.L., Miller, S.D.: Generating cryptographically-strong random lattice bases and recognizing rotations of $\mathbb{Z}^n$. CoRR (2021)
7. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC, pp. 575–584 (2013)
8. Chor, B., Rivest, R.L.: A knapsack-type public key cryptosystem based on arithmetic in finite fields. IEEE Trans. Inf. Theory **34**(5), 901–909 (1988)
9. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups, vol. 290. Springer, Heidelberg (2013)
10. Damgård, I.: On σ-protocols. Lecture Notes, University of Aarhus, Department for Computer Science (2002)
11. Ducas, L., Pierrot, C.: Polynomial time bounded distance decoding near Minkowski's bound in discrete logarithm lattices. Des. Codes Crypt. **87**(8), 1737–1748 (2019). https://doi.org/10.1007/s10623-018-0573-3
12. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
13. 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 (2002). https://doi.org/10.1007/3-540-46035-7_20
14. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM (JACM) **38**(3), 690–728 (1991)
15. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
16. Haviv, I., Regev, O.: Hardness of the covering radius problem on lattices. In: IEEE Conference on Computational Complexity, pp. 145–158 (2006)
17. Haviv, I., Regev, O.: On the lattice isomorphism problem. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 391–404. SIAM (2014)
18. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. CRC Press, Boca Raton (2020)

19. Klein, P.N.: Finding the closest lattice vector when it's unusually close. In: SODA, pp. 937–941 (2000)
20. Lapiha, O.: Comparing lattice families for bounded distance decoding near Minkowski's bound. Cryptology ePrint Archive Report 2021/1052 (2021). https://ia.cr/2021/1052
21. Lenstra, A.K., Lenstra, H.W., Jr., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
22. Lenstra, H.W.: On the Chor-Rivest knapsack cryptosystem. J. Cryptol. **3**(3), 149–155 (1991). https://doi.org/10.1007/BF00196908
23. Lenstra, H.W., Silverberg, A.: Revisiting the Gentry-Szydlo algorithm. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 280–296. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_16
24. Li, Z., Ling, S., Xing, C., Yeo, S.L.: On the bounded distance decoding problem for lattices constructed and their cryptographic applications. IEEE Trans. Inf. Theory **66**(4), 2588–2598 (2020)
25. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Coding Thv **4244**, 114–116 (1978)
26. Merkle, R., Hellman, M.: Hiding information and signatures in trapdoor knapsacks. IEEE Trans. Inf. Theory **24**(5), 525–530 (1978)
27. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective. The Kluwer International Series in Engineering and Computer Science, vol. 671. Kluwer Academic Publishers, Boston (2002)
28. Micciancio, D., Nicolosi, A.: Efficient bounded distance decoders for Barnes-Wall lattices. In: 2008 IEEE International Symposium on Information Theory, pp. 2484–2488. IEEE (2008)
29. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007). https://doi.org/10.1137/S0097539705447360
30. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007). Preliminary version in FOCS 2004
31. Mook, E., Peikert, C.: Lattice (list) decoding near Minkowski's inequality. IEEE Trans. Inf. Theory **68**(2), 863–870 (2022)
32. Odlyzko, A.M.: The rise and fall of knapsack cryptosystems. In: Pomerance, C. (ed.) Cryptology and Computational Number Theory. Proceedings of Symposia in Applied Mathematics, vol. 42, pp. 75–88 (1990)
33. Okamoto, T., Tanaka, K., Uchiyama, S.: Quantum public-key cryptosystems. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 147–165. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_9
34. Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 80–97. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_5
35. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_8
36. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. **40**(6), 1803–1844 (2011)
37. Plesken, W., Pohst, M.: Constructing integral lattices with prescribed minimum. I. Math. Comput. **45**(171), 209–221 (1985)
38. Plesken, W., Souvignier, B.: Computing isometries of lattices. J. Symb. Comput. **24**(3–4), 327–334 (1997)

39. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 1–40 (2009). Preliminary version in STOC 2005
40. Schnorr, C.P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**, 201–224 (1987)
41. Sendrier, N.: Finding the permutation between equivalent linear codes: the support splitting algorithm. IEEE Trans. Inf. Theory **46**(4), 1193–1203 (2000)
42. Serre, J.P.: A Course in Arithmetic, vol. 7. Springer, Heidelberg (2012)
43. Shamir, A.: A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. In: 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), pp. 145–152. IEEE (1982)
44. Sikiric, M.D., Haensch, A., Voight, J., van Woerden, W.P.: A canonical form for positive definite matrices. In: ANTS XIV, p. 179 (2020)
45. Solé, P., Charnes, C., Martin, B.: A lattice-based McEliece scheme for encryption and signature. Electron. Notes Discrete Math. **6**, 402–411 (2001)
46. Szydlo, M.: Hypercubic lattice reduction and analysis of GGH and NTRU signatures. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 433–448. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_27