Tensor-based Trapdoors for CVP and their Application to Public Key Cryptography

(Extended Version)

Roger Fischlin and Jean-Pierre Seifert

Fachbereich Mathematik (AG 7.2)
Johann Wolfgang Goethe-Universität Frankfurt am Main
Postfach 111932
D-60054 Frankfurt/Main, Germany

{fischlin,seifert}@informatik.uni-frankfurt.de
http://www.mi.informatik.uni-frankfurt.de/

January 3, 2000

Abstract. We propose two trapdoors for the Closest-Vector-Problem in lattices (CVP) related to the lattice tensor product. Using these trapdoors we set up a lattice-based cryptosystem which resembles to the McEliece scheme.[1]

Keywords. Public Key Cryptosystem, Closest Vector Problem, Lattice Reduction, Trapdoor, McEliece

1 Introduction

Since the invention of public key cryptography in 1976 by Diffie and Hellman [DH76] security of most cryptosystems is based on the (assumed) hardness of factoring or computing discrete logarithms. Only a few schemes based on other problems remain unbroken. Among which there is the McEliece scheme [St95] based on the computational difficulty of decoding a random code. It is still a challenge to develop new public key cryptosystem originating from the hardness of non number-theoretic problems.

In a pioneer work Ajtai [A96] constructed an efficiently computable function which is hard to invert on the average if the underlying lattice problem is intractable in the worst-case. This result has inspired many researchers. Ajtai

 This paper was presented at the 7th IMA conference Cryptography and Coding which took place 20–22 December 1999 at the Royal Agricultural College, Cirencester, UK. The proceedings have been published as Springer Lecture Notes in Computer Science, vol. 1746, editor Michael Walker (pp. 244–257). See http://www.springer.de/comp/lncs/index.html.

himself and Dwork [AD97] designed a public key cryptosystem based on the worst-case hardness of a lattice problem. But Nguyen and Stern [NS98] show that breaking the Ajtai-Dwork cryptosystem is unlikely $\mathcal{NP}$-hard and for realistic choices of the parameters one may recover the private key from the public key.

Using the idea from the McEliece cryptosystem it is straightforward to set up a cryptosystem based on the hardness of the Closest-Vector-Problem (CVP). A message is encoded as a lattice point plus a small error vector. To decipher the encrypted message look for the closest lattice point eliminating the small error. The open problem (which we address in this paper) is to find a suitable trapdoor. For the general case the only known trapdoor is an obvious one: a strongly reduced lattice base. But:

- If we apply lattice reduction algorithms to compute a reduced base for a given lattice, then an adversary can do it, too.
- In general it is not known how to create strongly reduced bases.

At Crypto '97 Goldreich, Goldwasser and Halevi [GGH97] took a practical approach to design a CVP-based cryptosystem (GGH scheme). Based on experiments they restrict themselves to a class of lattices defined by rather simple reduced bases. But simple attacks on the secret key show some weakness in the security of the trapdoor [SF+97]. A detailed cryptanalysis is given by Nguyen [N99].

In this paper we propose two trapdoors for the Closest-Vector-Problem based on the tensor product. We build a kind of strongly reduced lattice base using the tensor product of low dimensional lattices. The construction resembles to iterated codes where one efficiently decodes the tensor product given decoding algorithms for the individual codes. The second idea also applies the tensor product of lattices but in a different way. Finding the nearby vector in one component lattice enables to solve a restricted closest vector problem for the tensor product which is used to hide this secret structure.

2 Lattices, Reduction and Closest-Vector-Problem

In this section we recall facts about lattices and lattice reduction. To simplify, we usually restrict ourselves to full dimensional lattices.

Definition 1 (Lattice). *Given an ordered set (matrix) $B := [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ of n linear independent column vectors in $\mathbb{R}^m$, the set of all integral linear combinations of the vectors*

$$L = \mathcal{L}(B) := \left\{ \sum_{i=1}^{n} t_i \boldsymbol{b}_i \;\middle|\; t_i \in \mathbb{Z} \right\} = \sum_{i=1}^{n} \mathbb{Z} \boldsymbol{b}_i$$

is called a lattice generated by the base B. Its dimension is $\dim L := n$ and if $n = m$ we call it a full dimensional lattice. The vectors L are called lattice points. A lattice $L_{\text{sub}} \subseteq L$ with $\dim L_{\text{sub}} = \dim L$ is a sublattice of L. Sublattices of $\mathbb{Z}^m$ are called integer lattices.

For example, $\mathbb{Z}^n$ is a lattice and

$$D_n := \left\{ \boldsymbol{v} \in \mathbb{Z}^n \;\middle|\; \sum\nolimits_{i=1}^{n} v_i \equiv 0 \pmod 2 \right\}$$

is a sublattice of $\mathbb{Z}^n$ with $[\mathbb{Z}^n : D_n] = 2$ as there are exactly two cosets representing the vectors where the sum of entries is even/odd. Another integer lattice originating from coding theory is E_n which can be written for $n \equiv 0 \pmod 4$ as

$$E_n = \left\{ \boldsymbol{v} \in \mathbb{Z}^n \;\middle|\; \sum\nolimits_{i=1}^{n} v_i \equiv 0 \pmod 4, \quad v_i \equiv v_{i+1} \pmod 2 \right\}.$$

We have $[\mathbb{Z}_n : E_n] = 2^2 \cdot 2^{n-1} = 2^{n+1}$. These lattices D_n and E_n are well-studied [CS88, Chapter 4].

There are several bases for a lattice. Multiplying a base vector with -1 or adding an integral multiple of another base vector does not change the generated lattice. Two bases B, B' generate the same lattice iff there is an unimodular matrix

$$U \in \mathrm{GL}_n(\mathbb{Z}) := \left\{ U \in \mathbb{Z}^{n \times n} \;\middle|\; \det U = \pm 1 \right\}$$

with $B' = BU$ (note that $\mathrm{GL}_n(\mathbb{Z})$ is a group).

Definition 2 (Reciprocal Base and Lattice). *If B is a base for the lattice $L \subseteq \mathbb{R}^m$, then the unique $m \times n$ matrix B^* with $B \cdot (B^*)^T = \mathrm{Id}_n$ is called the reciprocal (or dual) base to B. The lattice $L^* := \mathcal{L}(B^*)$ is called the reciprocal lattice to L.*

The relation between primal/dual lattice and the lattice determinant are independent of the chosen lattice base for the lattice L:

Definition 3 (Determinant). *The determinant $\det L$ of a lattice $L = \mathcal{L}(B)$ with base vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ is the n-dimensional volume of the fundamental parallelepiped $\sum_{i=1}^{n} [0,1)\boldsymbol{b}_i$ which equals $\sqrt{\det(BB^T)}$ and in case of a full dimensional lattice $|\det B|$.*

Obviously $\det \mathbb{Z}^n = 1$ and using $[L : L_{\mathrm{sub}}] = \det L_{\mathrm{sub}} / \det L$ one derives $\det D_n = 2$ and $\det E_n = 2^{n+1}$. For a full dimensional lattice $L \subseteq \mathbb{Z}^n$ we have $(\det L) \cdot \boldsymbol{e}_i \in L$ for $i = 1, 2, \ldots, n$ [DKT87]. Thus, given a lattice base $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ one derives a base "reduced modulo the lattice determinant" for the same lattice by iteratively adding integral multiples of $(\det L) \cdot \boldsymbol{e}_i \in L$ such that the n vectors are linear independent and their entries are in $[0, \det L]$.[2]

With a base $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ of a lattice L we associate the *Gram-Schmidt orthogonalization* $\widehat{\boldsymbol{b}}_1, \widehat{\boldsymbol{b}}_2, \ldots, \widehat{\boldsymbol{b}}_n$ which is computed together with the Gram-Schmidt coefficients $\mu_{i,j} := \langle \boldsymbol{b}_i, \widehat{\boldsymbol{b}}_j \rangle / \|\widehat{\boldsymbol{b}}_j\|$ for $i > j$ by the recursion $\widehat{\boldsymbol{b}}_1 := \boldsymbol{b}_1$

[2] We cannot simply reduce the base vectors entries modulo $\det L$ because then the resulting vectors may be linear dependent and do not generate the lattice (for example, take $L = 2\mathbb{Z}$, $\boldsymbol{b}_1 = 2$ and $\det L = 2$).

and

$$\widehat{\boldsymbol{b}}_i := \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \widehat{\boldsymbol{b}}_j \qquad \text{for } i = 2, 3, \ldots, n.$$

$\|\widehat{\boldsymbol{b}}_i\|$ is the height of the i^{th} base vector. Unless stated otherwise, we use the standard scalar product $\langle \cdot, \cdot \rangle$ and the corresponding Euclidean norm $\|\cdot\|$. Letting $\mu_{i,i} := 1$ and $\mu_{i,j} := 0$ for $i < j$ one gets the Gram-Schmidt decomposition

$$[\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n] = \left[\widehat{\boldsymbol{b}}_1, \widehat{\boldsymbol{b}}_2, \ldots, \widehat{\boldsymbol{b}}_n \right] \cdot [\mu_{i,j}]^{\mathsf{T}}.$$

Observe that $\det L = \prod_{i=1}^{n} \|\widehat{\boldsymbol{b}}_i\|$ and that in general the vectors $\widehat{\boldsymbol{b}}_1, \widehat{\boldsymbol{b}}_2, \ldots, \widehat{\boldsymbol{b}}_n$ do not generate the lattice L. Call π_i the orthogonal projection

$$\pi_i : \operatorname{span}\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\} \to \operatorname{span}\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_i\}^{\perp}.$$

We have $\pi_i(b_i) = \widehat{\boldsymbol{b}}_i$.

Definition 4 (Successive Minima). *The i^{th} successive minimum $\lambda_i(L)$ of a lattice $L \subseteq \mathbb{R}^m$ is the smallest $\rho > 0$ such that there are i linear independent lattice points $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_i \in L \setminus \{0\}$ with $\|\boldsymbol{v}_i\| \leq \rho$.*

For example, $\lambda_1(\mathbb{Z}^n) = 1$, $\lambda(D_n) = \sqrt{2}$ and $\lambda_1(E_n) = \sqrt{8}$ for $n \geq 8$ [CS88]. Minkowski derived a general upper bound for the first successive minimum in terms of the lattice determinant [Ca97]:

Proposition 1 (Minkowski 1896). *If $L \subseteq \mathbb{R}^m$ is a lattice, then*

$$\lambda_1(L) \leq \sqrt{m} \cdot (\det L)^{\frac{1}{\dim L}}.$$

If we scale the lattice by a factor $\sigma > 0$, i.e. multiply each lattice point by σ, the successive minimum are scaled, too. To normalize the quantity $\lambda_1(L)$ one takes the ratio $\lambda_1(L)/\det L^{\frac{1}{\dim L}}$. The squared maximum of this ratio for full dimensional lattices is called the Hermite constant γ_n. According to the previous proposition we have $\gamma_n \leq n$.

The closest lattice point is uniquely determined if the minimal distance is less than $\frac{1}{2}\lambda_1(L)$ because otherwise the difference of two distinct nearby vectors would be a non-zero lattice point with length less than $\lambda_1(L)$. Given the value for the lattice determinant we look for lattices with large first successive minimum (so called dense lattices).

Definition 5 (Closest-Vector-Problem CVP). *Given a full dimensional lattice $L \subseteq \mathbb{R}^n$ and a point $\boldsymbol{x} \in \mathbb{R}^n$ the Closest-Vector-Problem is to find a lattice point $\boldsymbol{b} \in L$ with minimal distance $\mu(L, \boldsymbol{x}) := \min_{b \in L} \|\boldsymbol{x} - \boldsymbol{b}\|$.*

CVP is $\mathcal{NP}$-hard [Boas81,K87] and for large dimension it is conjectured to be "average-case" intractable. On the other hand, Babai [B86] proposed a procedure which efficiently approximates the nearby vector within a factor of $2^{n/2}$. Using a (fairly theoretical) algorithm of Schnorr [S87] one approximates in polynomial

time the closest vector up to a factor $(1+\epsilon)^n$ for any fixed $\epsilon > 0$, but the running time badly depends on ϵ. If the distance $\mu(L, \boldsymbol{x})$ is below a threshold and we know a suitable (reduced) base B for the lattice L, then CVP can be easily solved [B86,FK89]:

Lemma 1 (Nearest Plane). *Suppose L is a full dimensional lattice given by a base $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ with $\|\widehat{\boldsymbol{b}}_i\| > 2d$. For $\boldsymbol{x} \in \mathbb{R}^n$ with $\mu(L, \boldsymbol{x}) \leq d$ one can efficiently compute the uniquely determined closest lattice vector.*

The aim of lattice reduction is to find a base such that the vectors are rather orthogonal and the heights are large. We define lattice reduction in terms of the generalized β-reduction introduced by Schnorr [S87,S94]:

Definition 6 (Lattice Reduction). *Given a base $B = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ and $\beta \in [2, n]$ denote $L_{i,\beta} := \mathcal{L}(\boldsymbol{b}_i, \boldsymbol{b}_{i+1}, \ldots, \boldsymbol{b}_{\min(i+\beta-1,n)})$. We call the base B β-reduced if*

a) $|\mu_{i,j}| \leq \frac{1}{2}$ *for* $1 \leq j < i \leq n$.
b) $\|\widehat{\boldsymbol{b}}_i\| = \lambda_1(\pi_i(L_{i,\beta}))$ *for* $i = 1, 2, \ldots, n$.

There are two special cases: For $\beta = 2$ it is called LLL base and for $\beta = n$ one calls it HKZ base. B is a reciprocal β-reduced base, if the reduction holds for the reverse ordered reciprocal base B^* [LLS90].

For reduced bases in the sense of Hermite, Korkine and Zolotarev (HKZ) the first base vector is a shortest non-zero vector of the lattice. Like finding a shortest non-zero vector (Shortest Vector Problem, SVP) computing a HKZ base is intractable. To the best of our knowledge it is an open problem if a β-reduced base can be efficiently computed for a given lattice (even if β is fixed). For practical purposes an implementation of Schnorr and Hörner [SH95] quickly computes a β-reduced base for $\beta \leq 50$ and $n \leq 200$. On the other hand a reduced base in the sense of Lenstra, Lenstra and Lovász (LLL) can be computed in polynomial time [LLL82]. We have the following bounds for the heights of reduced bases:

Proposition 2 ([LLL82,LLS90]). *Let $B = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ be a base of the lattice $L \subseteq \mathbb{R}^m$. For $i = 1, 2, \ldots, n$*

a) If B is LLL reduced, then $\|\widehat{\boldsymbol{b}}_i\| \geq \dfrac{\lambda_i(L)}{2^{(i-1)/2}} \geq \dfrac{\lambda_1(L)}{2^{(n-1)/2}}$.

b) If B is reciprocal HKZ reduced, then $\|\widehat{\boldsymbol{b}}_i\| \geq \dfrac{\lambda_1(L)}{\gamma_i} \geq \dfrac{\lambda_1(L)}{n}$.

For reciprocal β-reduced bases the lower bound is about $\lambda_1(L)/\gamma_i \cdot \gamma_\beta^{n/\beta}$ (combine the proof of [LLS90, Prop. 4.1] and [S94, Theorem 4]).

3 Tensor Product of Lattices

Starting in 1954 when P. Elias introduced so called iterated or product code the tensor product has become a major way to combine two codes [MS77]. Given two

error correcting codes C_1, C_2 with generator matrices G_1, G_2 the iterated code C is the tensor product $C_1 \otimes C_2$ which is generated by the so called *Kronecker product* (direct product) $G_1 \otimes G_2$ of both generator matrices. The minimal distance $d(C)$ equals to $d(C_1) \cdot d(C_2)$, i.e. the product of the minimal distances of the component codes.

The tensor product applies to lattices, too [M96]. Given two full dimensional lattice bases A, B the Kronecker product $A \otimes B$ of the two matrices is a base for the tensor product $\mathcal{L}(A) \otimes \mathcal{L}(B)$ which is a lattice, too. The Kronecker product of a $k \times \ell$ matrix $A = [a_{ij}]$ and an $m \times n$ matrix $B = [b_{ij}]$ is the $km \times \ell n$ matrix obtained from A by replacing each entry a_{ij} by $a_{ij}B$.

$$A = \begin{bmatrix} a_{11} & \dots & a_{1\ell} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{k\ell} \end{bmatrix} \qquad A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1\ell}B \\ \vdots & \ddots & \vdots \\ a_{k1}B & \dots & a_{k\ell}B \end{bmatrix}$$

For example, scaling a lattice L by a factor $\sigma > 0$ can be written as tensor product $\sigma L = L \otimes \sigma\mathbb{Z}$ as $[\sigma]$ is a base of $\sigma\mathbb{Z}$. Despite the fact that in general $A \otimes B \neq B \otimes A$ the associative and distributive law still hold [L96]:

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C$$
$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

Let $L := L_1 \otimes L_2$ denote the tensor product of two full-dimensional lattices $L_1 \subseteq \mathbb{R}^{n_1}$ and $L_2 \subseteq \mathbb{R}^{n_2}$. The lattice point $\boldsymbol{b} := (b_1, b_2, \dots, b_{n_1 n_2}) \in L$ can be written as a two dimensional array

$$\begin{array}{llll} b_1 & b_2 & \dots\ b_{n_2} & \in L_2 \\ b_{n_2+1} & b_{n_2+2} & \dots\ b_{2n_2} & \in L_2 \\ \vdots & \vdots & \vdots & \vdots \\ b_{(n_1-1)n_2+1} & b_{(n_1-1)n_2+2} & \dots\ b_{n_1 n_2} & \in L_2 \\ \in & \in & \in & \\ L_1 & L_1 & L_1 & \end{array}$$

such that the column vectors belong to the lattice L_1 and the row vectors to L_2. The converse is true for product codes but not for lattices: for example, $2 \in 2\mathbb{Z}$ but $2 \notin 2\mathbb{Z} \otimes 2\mathbb{Z} = 4\mathbb{Z}$.

Finally, let us recall some facts about the tensor product of lattices, for details see [K93, Chapter 7] or [M96, §1.10]. It is easy to verify [M96, Prop. 10.1]:

Proposition 3. *Suppose L_1, L_2 are two full dimensional lattices. Then*

$$\dim(L_1 \otimes L_2) = \dim L_1 \cdot \dim L_2$$
$$\det(L_1 \otimes L_2) = (\det L_1)^{\dim L_2} \cdot (\det L_2)^{\dim L_1}.$$

Using induction we derive for the tensor product of t lattices $L_1, \dots, L_t$:

$$\det(L_1 \otimes \dots \otimes L_t) = \prod_{i=1}^{t} \det L_i^{\prod_{j \neq i} \dim L_j}. \tag{1}$$

For the first successive minimum we have the following result given in Lemma 7.1.1 and Theorem 7.1.1 of [K93]. The bound 43 is related to the Hermite constant γ_n for which only lower and upper bounds are known (except for $n \leq 8$).

Proposition 4. *Suppose L_1, L_2 are two full dimensional lattices. Then*

$$\lambda_1(L_1 \otimes L_2) \leq \lambda_1(L_1) \cdot \lambda_1(L_2)$$

with equality if $\dim L_1 \leq 43$ *or* $\dim L_2 \leq 43$.

For any dimension $n \geq 292$ examples with non-equality are known. Compared to the tensor product of two codes this differs as the minimal distance of iterated codes always equals to the product of the minimal distances.

4 CVP-based Public Key Cryptosystems

We describe the general frame work for public key cryptosystems based on the computational hardness of CVP and present a general attack on these schemes. We compare these systems with the well-known analogous McEliece scheme based on error correcting codes.

Frame Work. We call lattices $L \subseteq \mathbb{R}^n$ $d(n)$-decodable if using a trapdoor one can easily determine the nearby vector for $\boldsymbol{x} \in \mathbb{R}^n$ with $\mu(L, \boldsymbol{x}) \leq d(n)$ (bounded distance decoding). Let B be a base of the $d(n)$-decodable lattice L. The trapdoor (for example B) is the secret key. The public key consists of a base B_{pub} for the public lattice $L_{\text{pub}} := \mathcal{L}(B_{\text{pub}})$ which is related to L (or even equals L). By choosing a public base B_{pub} one must regard two aspects:

1. Given the base B_{pub} an adversary may not get "useful information" about the secret trapdoor enabling him to break the system.
2. It should be intractable to solve the closest vector problem by simply applying lattice reduction to B_{pub}.

To avoid native attacks select a random unimodular matrix U transforming the base B. Analogous to the McEliece scheme we use a random orthogonal mapping R (rotation, i.e. $R^{-1} = R^{\mathsf{T}}$) to hide the trapdoor (tensor lattice structure). Set

$$B_{\text{pub}} := R \cdot (BU).$$

Note that rotating the base also changes the lattice, i.e. $L_{\text{pub}} = R(L)$, meanwhile this does not change the vector lengths nor the lattice determinant. To encrypt a message $\boldsymbol{m} \in \mathbb{Z}^n$ select an error vector $\boldsymbol{e} \in \mathbb{R}^n$ with $\|\boldsymbol{e}\| \leq d(n)$ and send

$$\boldsymbol{y} := B_{\text{pub}}\boldsymbol{m} + \boldsymbol{e} = RBU\boldsymbol{m} + \boldsymbol{e}.$$

Using the trapdoor one determines the closest vector to $R^{-1}\boldsymbol{y} = B(U\boldsymbol{m}) + R^{-1}\boldsymbol{e}$ which is $\boldsymbol{b} := U\boldsymbol{m}$ because $\left\|R^{-1}\boldsymbol{e}\right\| = \|\boldsymbol{e}\| \leq d(n)$. Now $U^{-1}\boldsymbol{b}$ equals the plaintext $\boldsymbol{m}$.

Choosing the unimodular matrix U is done by multiplying several elementary matrices (representing the elementary operations: exchanging two columns, adding a multiple of a column to another, flipping the sign of a column) [GGH97]. Choosing a random rotation R is more difficult. This problem has been addressed by Sloane [Sl82] suggesting orthogonal matrices based on Hadamard matrices. This solves a second problem as, in general, orthogonal matrices have real coefficients. But for the restricted class of orthogonal matrices the elements are rational numbers such that scaling the lattice by $\sqrt{n}$ yields an integer one (details are given in Appendix A). Then we can reduce the public base "modulo $\det L_{\text{pub}}$".

But scaling has a disadvantage: The determinant increases by a factor $(\sqrt{n})^n$ and the bit length of the public key grows by $\mathcal{O}\big(n^3 \log_2 n\big)$ bits. So depending on the trapdoor and efficiency one may use only a permutation matrix P instead of a rotation R. Or simply apply a unimodular transformation without any rotation. The GGH scheme just applies a permutation as the structure of the secret base is publicly known. It uses as secret key a random lattice base

$$B \in_{\mathrm{R}} \sqrt{n} \cdot \mathrm{Id}_n + [-4, +4]^{n \times n}$$

which (based on experiments) enables one to decrypt a message with probability of order $1 - \frac{1}{n}$ for a random error vector $\boldsymbol{e} \in_{\mathrm{R}} \{\pm\sigma\}^n$ where $\sigma = 2$.

Finding the Tensor Decomposition. To the best of our knowledge no efficient algorithm for finding a tensor lattice decomposition is known. For matrices over a finite field a tensor decomposition can be found in (small) dimensions [OL97]. But for the CVP trapdoor the matrices are over $\mathbb{Z}$, the dimension is rather large and the coordinates are permutated (respectively we rotate the lattice).

One possible weakness is the special form of the lattice determinant (1). Factoring this number may yield the dimension of the component lattices. To counteract this possible weakness simply choose lattices with equal determinant whose dimensions have many prime factors (for example, powers of 2). For a small number of component lattices (say $t = 2$) one might alternatively select lattices with determinants having many prime factors in a way that factoring (1) permits many decompositions. Even if the adversary cannot deduce the dimensions from the determinant he knows that the dimensions of the component lattices are at most 43. In case of a small number of component lattices the adversary might try all possible decompositions of the dimension.

To undo the permutation P the adversary tries to identify coordinates belonging to the same copy of a component lattice. Let $L_1 = \mathcal{L}(A)$ and $L_2 = \mathcal{L}(B)$ denote the component lattices and $L := P(L_1 \otimes L_2)$ the public key. From the proof given in Appendix B we derive

$$\det L = \prod_{i=1}^{\dim L_1} \prod_{j=1}^{\dim L_2} \|\widehat{\boldsymbol{a}}_i\| \cdot \|\widehat{\boldsymbol{b}}_j\| = \prod_{i=1}^{\dim L_1} \|\widehat{\boldsymbol{a}}_i\|^{\dim L_2} \det L_2.$$

If one removes a coordinate then the heights change, the new lattice has no tensor structure and the determinant is no longer of the form (1). But if we delete $\dim L_2$ coordinates belonging to the same i^{th} copy of the component lattice L_2, then the lattice determinant is divided by $\|\widehat{\boldsymbol{a}}_i\|^{\dim L_2} \det L_2$. But in general this in no integer because the height is a real number (although its square is a rational number). So the adversary cannot verify the correctness of his choice by checking the divisibility. For higher dimension this "attack" is intractable as there are $\binom{\dim L_1 \cdot \dim L_2}{\dim L_2}$ possible choices.

Attacks by Lattice Reduction. There is a simple but very powerful heuristic attack on the GGH scheme and all other CVP-based cryptosystems [SF+97]. Given an encrypted message $\boldsymbol{x} = B_{\text{pub}}\boldsymbol{m} + \boldsymbol{e}$ apply a lattice reduction algorithm to the lattice L_{ext} generated by the $(n+1) \times (n+1)$ matrix

$$B_{\text{ext}} := \begin{bmatrix} \boldsymbol{x} & B_{\text{pub}} \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e} & B_{\text{pub}} \\ 1 & 0 \end{bmatrix} \cdot U, \qquad \text{for some } U \in \mathrm{GL}_{n+1}(\mathbb{Z}).$$

Experiments suggest that the shortest non-zero vector of L_{ext} is $\boldsymbol{e}_{\text{ext}} := \pm(\boldsymbol{e}, 1)$ (which yields the plaintext) and that $\lambda_2(L_{\text{ext}}) = \lambda_1(L_{\text{pub}})$. To retrieve $\boldsymbol{e}_{\text{ext}}$ the adversary approximates the shortest vector for L_{ext} within a factor less than $\lambda_1(L_{\text{pub}})/\|\boldsymbol{e}\|$ (ignoring the additional ± 1 entry).[3] To minimize this ratio take $\|\boldsymbol{e}\| \approx \frac{1}{2}\lambda_1(L)$ such that the adversary retrieves the message if he computes the shortest non-zero vector up to a factor of ≈ 2. The LLL algorithm [LLL82] approximates the shortest vector up to a factor of $2^{\frac{n-1}{2}}$ and for β-reduced bases the factor is about $2^{n/\beta}$ [S94, Theorem 4].

Although in case of the GGH scheme one must approximate the shortest vector within a factor 2, in dimension 150 it takes 15 minutes to retrieve the error vector [SF+97]. As within $1\frac{1}{2}$ hours lattice reduction even yields the secret key for the lattice. Thus, it is questionable if the class of lattice is a good choice. The recent cryptanalysis of Nguyen [N99] strengthens these results and reveals a second weakness: as $\boldsymbol{e} \in \{\pm\sigma\}^n$ the adversary retrieves $\boldsymbol{m} \bmod 2\sigma$ by solving the modular system

$$\boldsymbol{y} + [\sigma, \ldots, \sigma]^T = B_{\text{pub}}\boldsymbol{m} \pmod{2\sigma}$$

which simplifies the closest vector problem where the error vector length is now $\frac{1}{2}\sqrt{n}$ compared to $\sigma\sqrt{n}$. To counteract this attack the error vector should not have a pattern like $\boldsymbol{e} \in \{\pm\sigma\}^n$. For example, take a random $\boldsymbol{e} \in_{\text{R}} [-\sigma, +\sigma]^n$.

Comparison with McEliece Scheme. Although being one of the first public key cryptosystems and still remaining unbroken the McEliece scheme has not become widely used because rather large key is required. It is commonly believed that CVP-based cryptosystems are weaker than the analogous McEliece schemes

[3] We say an algorithm approximates the shortest vector for the lattice L within a factor $\tau \geq 1$ if it outputs $\boldsymbol{b} \in L \setminus \{0\}$ with $\|\boldsymbol{b}\| \leq \tau \cdot \lambda_1(L)$.

due to the powerful lattice reduction algorithms. On the other hand, the strong attacks on the McEliece schemes [CSe98] seem not to be suitable for its lattice based variants. Let G be the generating $n \times k$ matrix of the secret $[n, k, d]$ error correcting code C, i.e. $C \subseteq \{0,1\}^n$, $\dim C = k$ and the minimal Hamming distance is d. The public matrix for the McEliece scheme is

$$G_{\text{pub}} := SGP$$

over the field $\mathbb{Z}/2\mathbb{Z}$ where $S \in \mathrm{GL}_k(\mathbb{Z}/2\mathbb{Z})$ and P is a permutation matrix. The matrix S ensures that the generating matrix G is not systematic, e.g. $G \neq [\mathrm{Id}_k \ A]$, because otherwise an encrypted message $\boldsymbol{y} := \boldsymbol{m}G_{\text{pub}} + \boldsymbol{r}$ reveals some bits of the plain text. The trapdoor (the error correcting code) is only hidden by applying the permutation which restricts the class of suitable codes.

For codes and lattices multiplying the generating matrix respectively base matrix by an unimodular matrix does not change the code and lattice. Applying a permutation P changes the code into an equivalent code while the hamming weight of code words remains. Applying a rotation R changes the lattice into an isometric lattice while the Euclidean length of lattice vectors remains. In both cases knowing the transformation one reduces decoding respectively find the closest vector to the original code (lattice). But without this knowledge it is assumed that an adversary does not get any "useful information" about the underlying structure.

Although there are unique normal forms for lattice bases (like the Hermite normal form [DKT87,Co93]) they do not play the same role as systematic generator matrices for codes where most algorithms are based on this form. The well-known normal forms for lattice bases rely on the matrix structure rather on reduced lattice bases. Nevertheless the known SVP algorithms are far more powerful than algorithms for finding the smallest code word. Thus, it seems that the McEliece scheme is more liable to structural attacks trying to retrieve the secret key meanwhile for CVP systems it is easier to regain the error vector of a single message.

CVP-Trapdoor Based on Tower of Error Correcting Codes. Given a tower $C_t \subseteq C_{t-1} \subseteq \cdots \subseteq C_1 \subseteq \{0,1\}^n$ of (suitable)[4] binary nested error correcting codes one uses "Construction D" of Barnes and Sloane [CS88, Chapter 8.1] to create a dense lattice L. The closest vector is found using a bounded-distance multi-stage decoding algorithm based on the error correcting codes [F89]. Although we can find the unique closest vector up to the limit $\frac{1}{2}\lambda_1(L)$ it may be difficult to find a suitable tower. Lattices generated by the given examples [CS88] are well-known such that an adversary even might conclude the underlying choice from the public lattice base. As the choice for the tower is restricted in most cases we have $t = 1$. Then the lattice is $L = C + 2\mathbb{Z}^n$ for a binary code C (whose code words are written as elements of $\mathbb{Z}^n$) and the trapdoor equals

[4] The choice of the error correcting codes has to be restricted to ensure that L is actually a lattice.

the McEliece trapdoor. The construction can be relaxed [CP94,FV96] but these lattices have a higher dimension and are not as dense as for "Construction D".

5 HKZ-Tensor-Trapdoor

We construct a kind of strongly reduced lattice bases for a large general class of lattices using the tensor product to iteratively combine low dimensional lattices. If A and B are reciprocal HKZ-reduced then the lower bound for the heights of $A \otimes B$ is the product of both lower bounds. This enables us to apply Nearest-Plane like given a reciprocal HKZ base. In Appendix B we show:

Proposition 5. *Suppose A and B are bases of two lattices with $\|\widehat{\boldsymbol{a}}_i\| \geq h_A$ and $\|\widehat{\boldsymbol{b}}_i\| \geq h_B$. For the base $C := A \otimes B$ of $\mathcal{L}(A) \otimes \mathcal{L}(B)$ we have $\|\widehat{\boldsymbol{c}}_i\| \geq h_A h_B$.*

Using this result, one sets up the candidate trapdoor by composing the lattice L with dimension n out of t lattices $L_1, L_2, \ldots, L_t$ with $\dim L_i \in [2, 43]$. For simplicity we assume all lattices have the same dimension c, e.g. $n = c^t$ and $t = \log_c n$. As we do not require any secret structure for the underlying low dimensional lattices we take "random" lattices (see discussion in Appendix C). For these lattices of fixed and low dimension we can efficiently compute (reciprocal) HKZ bases both in a theoretical setting [K87] and in practice [SH95]. Let $B_1, B_2, \ldots, B_t$ denote the reciprocal HKZ bases,

$$B := B_1 \otimes B_2 \otimes \ldots \otimes B_t$$

their Kronecker product and $L := \mathcal{L}(B)$ the tensor product lattice. By induction based on Proposition 3 we derive

$$\det L = \prod_{i=1}^{t} (\det L_i)^{c^t} = \Big(\prod_{i=1}^{t} \det L_i \Big)^n$$

and according to Proposition 4 and Proposition 5:

$$\lambda(L) = \prod_{i=1}^{t} \lambda_1(L_i) \qquad\qquad \|\widehat{\boldsymbol{b}}_i\| \geq \prod_{i=1}^{t} \frac{\lambda_1(L_1)}{\gamma_c} = \frac{\lambda_1(L)}{\gamma_c^t}$$

Now applying Nearest-Plane we are in the same situation as given a reciprocal HKZ base for the lattice L. Using the Kronecker product of the reciprocal HKZ-reduced bases we find the nearby vector if the distance is below $\frac{1}{\gamma_c^t}\lambda_1(L)$. An adversary trying to find the error vector by embedding it into a SVP problem has to approximate the shortest lattice vector within a factor at least γ_c^t. Using stronger bounds [CS88, Chapter 1] we have $\gamma_c < 1.75 + 0.12c$ for $c \in [8, 50]$. For $n = 529$ this means: taking two 23-dimensional lattices (i.e. $c = 23$, $t = 2$), the adversary has to approximate the shortest vector within a factor less than 21. Note, this is a worst case scenario and by building the lattice one may simply re-select a small lattice L_i if the heights of the HKZ base are too close to $\frac{\lambda_1(L_i)}{\gamma_c}$.

6 Tensor-Hiding-Trapdoor

Again, we use the tensor product but unlike creating a reduced base for the product we try to hide the component lattices. We solve a restricted Closet-Vector-Problem for the tensor product using an algorithm which finds the nearby vector in a component lattice. To generate a candidate trapdoor for an n-dimensional lattice L select two full dimensional lattices L_{decode} and L_{hide} with the following properties:

- We have $n_{\text{decode}} := \dim L_{\text{decode}} \leq 43$, $n_{\text{hide}} := \dim L_{\text{hide}} = \frac{n}{n_{\text{decode}}}$.
- L_{decode} is a dense lattice, i.e., for a given lattice determinant the first successive minimum $\lambda_1(L_{\text{decode}})$ is large.
- For the lattice L_{decode} we efficiently find the nearby vector if the distance is at most a large threshold d_{decode}, for example $d_{\text{decode}} \approx \frac{1}{2}\lambda_1(L_{\text{decode}})$. Let $\sigma := d_{\text{decode}}/\sqrt{n_{\text{decode}}}$ denote the distances in terms of the maximum norm.
- We have for a small $\kappa \geq 1$:

$$\frac{\sqrt{2n_{\text{decode}}} \cdot d_{\text{decode}}}{\lambda_1(L_{\text{decode}})} \leq \lambda_1(L_{\text{hide}}) \leq \kappa \cdot \frac{\sqrt{2n_{\text{decode}}} \cdot d_{\text{decode}}}{\lambda_1(L_{\text{decode}})}$$

  The lower bound ensures that the plaintext still corresponds to the closest vector and the upper bound makes it intractable finding the message by means of simple lattice reduction.

The lattice L_{decode} may be based on a tower of nested error correcting codes. For the lattice L_{hide} use the tensor product and (for example) the lattices D_n, E_n or the method given in Appendix C Let $L := L_{\text{decode}} \otimes L_{\text{hide}}$ (or visa versa) denote the public lattice. As in Section 4 we set up the cryptosystem but with two restrictions:

1. The error vector is $\boldsymbol{e} \in [-\sigma, +\sigma]^n$ (note $\|\boldsymbol{e}\| \leq \frac{1}{2}\sqrt{2n} \cdot d_{\text{decode}}$) and
2. Instead of using an orthogonal matrix we just apply a permutation because rotations do not not keep the length in terms of the maximum norm fixed.

According to the choice of $\lambda_1(L_{\text{hide}})$ and $n_{\text{decode}} \cdot n_{\text{hide}} = n$ Proposition 4 implies

$$\sqrt{2n} \cdot d_{\text{decode}} \leq \lambda_1(L) \leq \kappa \cdot \sqrt{2n} \cdot d_{\text{decode}}.$$

Retrieving the paintext is done by computing the unique nearby point: given a point $\boldsymbol{y} := \boldsymbol{b} + \boldsymbol{e}$ with $\boldsymbol{b} \in L$ we write $\boldsymbol{y}$ as the two-dimensional array and apply the given algorithm to determine the nearby point in L_{decode} for the $\dim L_{\text{hide}}$ rows. The concatenation of the results is the closest vector in L because otherwise there are two nearby lattice points. On the other hand, an adversary trying to find the error vector by embedding it into a SVP problem has to approximate the shortest lattice vector within a factor of at least 2κ. To avoid lattice reduction attacks the lattice dimension should be at least 500.

7 Conclusions and Open Problems

We have suggested two trapdoors for the Closest-Vector-Problem based on the conjectured computational difficulty of finding tensor decomposition after applying a permutation (respectively a rotation). It is nearly as hard as possible to retrieve the cipher text by means of simply applying lattice reduction such that these "brute force" attacks should not succeed. But like the McEliece scheme the public key is very large in a way that CVP-based crypto systems have no practical impact as long as factoring or discrete logarithm remain intractable.

Given two lattices L_1, L_2 and oracles solving the closest vector problem in L_1, L_2, can one efficiently compute the nearby vector for the tensor product $L_1 \otimes L_2$? To the best of our knowledge this is an open problem whereas iterated codes are majority decodable if one of the component codes is majority decodable [R70]. Can one take advantage from these decoding algorithm? One obstacle may be that for tensor lattices the lattice vectors cannot be characterized by a two dimensional array as opposite to product codes.

8 Acknowledgment

We thank Marc Fischlin for proof reading.

References

[A96] M. Ajtai: *Generating Hard Instances of Lattice Problems*, in Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), ACM Press, pp. 402–412, 1996.

[AD97] M. Ajtai and C. Dwork: *A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence*, in Proceedings of the 29$^{\text{th}}$ Annual ACM Symposium on Theory of Computing (STOC), ACM Press, pp. 284–293, 1997.

[B86] L. Babai: *On Lovász' Lattice Reduction and the Nearest Lattice Point Problem*, Combinatorica, vol. 6, pp. 1–13, 1986.

[CSe98] A. Canteaut and N. Sendrier: *Cryptanalysis of the Original McEliece Cryptosystem*, Advances in Cryptology — Proceedings Asiacrypt '98, Lecture Notes in Computer Science, vol. 1541, Springer Verlag, Berlin/Heidelberg, pp. 187-199, 1998.

[Ca97] J.W.S. Cassels: *An Introduction to the Geometry of Numbers*, Springer Verlag, Berlin/Heidelberg, 1997.

[Co93] H. Cohen: *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics, vol. 138, Springer Verlag, Berlin/Heidelberg, 1993.

[CS88] J.H. Conway and N.J. Sloane: *Sphere Packings, Lattices and Groups*, Springer Verlag, New York, 1988.

[CP94] M.A.O. da Costa e Silva and R. Palazzo, Jr.: *A Bounded-Distance Algorithm for Lattices Obtained from a Generalized Code Formula*, IEEE Transaction on Information Theory, Vol. 40(6), pp. 2075–2082, 1994.

[DKT87] P.D. Domich, R. Kannan and L.E. Trotter: *Hermite Normal Form Computation using modulo Determinant Arithmetic*, Mathematics of Operation Research, vol. 12(1), pp. 50–59, 1987.

[DH76] W. Diffie and M. Hellman: *New Directions in Cryptography*, IEEE Transaction on Information Theory, Vol. 22(6), pp. 644–654, 1976.

[D97] C. Dwork: *Positive Applications of Lattices to Cryptography*, Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, vol. 1295, Springer Verlag, Berlin/Heidelberg, pp. 44–51, 1997.

[Boas81] P. van Emde Boas: *Another $\mathcal{NP}$-complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice*, technical report 81-04, Department of Mathematics, University of Amsterdam, 1981.

[FK89] M.L. Furst and R. Kannan: *Succinct Certificates for Almost all Subset Sum Problems*, SIAM Journal on Computing, vol. 18(3), pp. 550–558, 1989.

[F89] G.D. Forney, Jr.: *A Bounded-Distance Algorithm for the Leech Lattices with Generalization*, IEEE Transaction on Information Theory, Vol. 35(4), pp. 960–909, 1989.

[FV96] G.D. Forney, Jr. and A. Vardy: *Generalized Minimum Distance Decoding of Euclidean-space Codes and Lattices*, IEEE Transaction on Information Theory, Vol. 42, pp. 1992–2026, 1996.

[GGH97] O. Goldreich, S. Goldwasser and S. Halevi: *Public-Key Cryptosystems from Lattice Reduction Problems*, Advances in Cryptology — Proceedings Crypto '97, Lecture Notes in Computer Science, vol. 1294, Springer Verlag, Berlin/Heidelberg, pp. 112–131, 1997. Previous version as ECCC report TR96-056.

[K87] R. Kannan: *Minkowski's Convex Body Theorem and Integer Programming*, Mathematics of Operation Research, vol. 12(3), pp. 415–440, 1987.

[K93] Y. Kitaoka: *Arithmetic of Quadratic Forms*, Cambridge Tracts in Mathematics, vol. 106, Cambridge University Press, Cambridge, 1993.

[LLL82] A.K. Lenstra, H.W. Lenstra and L. Lovász: *Factoring Polynomials with Rational Coefficients*, Springer Mathematische Annalen, vol. 261, pp. 515–534, 1982.

[LLS90] J.C. Lagarias, H.W. Lenstra and C.P. Schnorr: *Korkin-Zolotarev Bases and successive Minima of a Lattice and its Reciprocal Lattice*, Combinatorica, vol. 10, pp. 333–348, 1990.

[L96] H. Lütkepohl: *Handbook of Matrices*, John Wiley & Son, Chichester, England, 1996.

[MS77] F.J. MacWilliams and N.J. Sloane: *The Theory of Error Correcting Codes*, North-Holland Mathematical Library Vol. 16, North-Holland, Amsterdam, 1977.

[M96] J. Martinet: *Les Réseaux Parfaits des Espaces Euclidiens*, Masson, Paris, 1996.

[MO90] J.E. Mazo and A.M. Odlyzko: *Lattice Points in high-dimensional Spheres*, Monatshefte Mathematik, vol. 110(1), pp. 47–61, 1990.

[NS98] P. Nguyen and J. Stern: *Cryptanalysis of the Ajtai-Dwork Cryptosystem*, Advances in Cryptology — Proceedings Crypto '98, Lecture Notes in Computer Science, vol. 1294, Springer Verlag, Berlin/Heidelberg, pp. 223–242, 1998.

[N99] P. Nguyen: *Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97*, Advances in Cryptology — Proceedings Crypto '99, Lecture Notes in Computer Science, vol. 1666, Springer Verlag, Berlin/Heidelberg, pp. 288–304, 1999.

[OL97] E. O'Brien and C.R. Leedham-Green: *Recognising Tensor Products of Matrix Groups*, International Journal Algebra Computing, vol. 7, pp. 541–559, 1997.

[PS87] A. Paz and C.P. Schnorr: *Approximating Integer Lattices by Lattices with cyclic Factor Group*, in Proceedings of the 14.th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, vol. 267, Springer Verlag, Berlin/Heidelberg, pp. 386–393, 1987.

[R70] S.M. Reddy: *On Decoding Iterated Codes*, IEEE Transaction on Information Theory, Vol. 16(5), pp. 624–627, 1970.

[S87] C.P. Schnorr: *A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms*, Theoretical Computer Science, vol. 53, pp. 201–224, 1987.

[S94] C.P. Schnorr: *Block Reduced Lattice Bases and Successive Minima*, Combinatorics, Probability and Computing, vol. 3, pp. 507–522, 1994.

[SH95] C.P. Schnorr and H.H. Hörner: *Attacking the Chor-Rivest Cryptosystem by improved Lattice Reduction*, Advances in Cryptology — Proceedings Eurocrypt '95, Lecture Notes in Computer Science, vol. 921, Springer Verlag, Berlin/Heidelberg, pp. 1–12, 1995.

[SF+97] C.P. Schnorr, M. Fischlin, R. Fischlin, H. Koy and A. May: *Lattice Attacks on the GGH Cryptosystem*, Crypto '97 Rump Session, 1997.

[Sl82] N.J.A. Sloane: *Encryption by Random Rotations*, in Proceedings of the Workshop on Cryptography Burg Feuerstein, 1982, Lecture Notes in Computer Science, vol. 149, Springer Verlag, Berlin/Heidelberg, pp. 71–128, 1983.

[St95] D.R. Stinson: *Cryptography: Theory and Practice*, CRC Press, Boca Raton, 1995.

A Quasi-random Orthogonal Matrix

In Section 4 we faced the problem how to generate a random orthogonal matrix and how to transform the possible real lattice into an integer one. Sloane [Sl82] has suggested using orthogonal matrices based on Hadamard matrices. He calls the generated rotations quasi-random orthogonal matrices. Let H_n be a Hadamard matrix of order n, in other words H_n is an $n \times n$ matrix with coefficients ± 1 and $H_n H_n^{\mathsf{T}} = n\mathrm{Id}_n$. Note that $\frac{1}{\sqrt{n}} H_n$ is an orthogonal matrix. Flipping the sign of any row or column changes a Hadamard matrix into another. There are several known constructions for Hadamard matrices if the order n is a multiple of 4 [MS77, Ch. 2, §3]. For example, if n is a power of 2 one gets a Hadamard matrices by the recursion $H_1 = [1]$ and

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}.$$

To create an orthogonal matrix choose arbitrary permutations matrices P, P' and diagonal matrices D, D' with coefficients ± 1. Let

$$R := \frac{1}{\sqrt{n}} \cdot DPH_nP'D'$$

i.e. we permute and flip the sign of the rows and columns of the Hadamard matrix. Restrict n to powers of 2 and scale the lattice to get integral coefficients:

$$B_{\text{pub}} := \sqrt{n} \cdot RBU = (DPH_nP'D')BU$$

As we scale the lattice in each direction we have $\mathcal{L}(B_{\text{pub}}) = \sqrt{n} \cdot \mathcal{L}(RB)$. Successive minima and the base heights are also stretched by $\sqrt{n}$ and instead of $\boldsymbol{e}$ with $\|\boldsymbol{e}\| \leq d$ we demand $\|\boldsymbol{e}\| \leq \sqrt{n}d$. Clearly, the complexity of the closest vector problems remains the same. But the determinant increases by $\sqrt{n}^{\,n}$. To reduce this disadvantage may combine independent rotations which apply only to a fixed number c of directions. Select $c \times c$ random orthogonal matrices $R_1, R_2, \ldots, R_{\frac{n}{c}}$ and a $n \times n$ permutation matrix P':

$$R := P' \cdot \begin{bmatrix} R_1 & & & \\ & R_2 & & 0 \\ & & \ddots & \\ 0 & & & R_{\frac{n}{c}} \end{bmatrix} \cdot P'.$$

Using this class of orthogonal matrices it is sufficient to scale the lattice by a factor $\sqrt{c}$ instead of $\sqrt{n}$.

B Proof of Proposition 5

Given two bases A, B of lattices with $\|\widehat{\boldsymbol{a}}_i\| \geq h_A$ and $\|\widehat{\boldsymbol{b}}_i\| \geq h_B$ we show $\|\widehat{\boldsymbol{c}}_i\| \geq h_A h_B$ for the base $C := A \otimes B$ of $\mathcal{L}(A) \otimes \mathcal{L}(B)$. Recall a well-known result [L96]:

Proposition 6. *For two orthogonal matrices A, B the Kronecker product $C := A \otimes B$ is orthogonal, too.*

We use the so called Iwasawa decomposition [Co93, Corollary 2.5.6]. The base matrix C can be uniquely written as $C = DOT$ with

- a diagonal matrix D,
- an orthogonal matrix O and
- an upper triangle matrix T with diagonal elements equal to 1.

Compare the Iwasawa decomposition to the Gram-Schmidt decomposition. As $\widehat{\boldsymbol{c}}_1, \widehat{\boldsymbol{c}}_2, \ldots, \widehat{\boldsymbol{c}}_n$ are orthogonal we have that T is the transposed Gram-Schmidt matrix and the i^{th} diagonal entry of D equals the height $\|\widehat{\boldsymbol{c}}_i\|$. Let $A = D_A O_A T_A$ and $B = D_B O_B T_B$ denote the Iwasawa decomposition of A and B. Using the associative and distributive law [L96]

$$U \otimes (X \otimes Y) = (U \otimes X) \otimes Y$$
$$(U \otimes V)(X \otimes Y) = (UX) \otimes (VY)$$

we get for the Kronecker product $C = A \otimes B$:

$$\begin{aligned} C &= \big(D_A O_A T_A\big) \otimes \big(D_B O_B T_B\big) \\ &= \big([D_A O_A] \cdot T_A\big) \otimes \big([D_B O_B] \cdot T_B\big) \\ &= \big([D_A O_A] \otimes [D_B O_B]\big) \cdot \big(T_A \otimes T_B\big) \\ &= \big(D_A \otimes D_B\big) \cdot \big(O_A \otimes O_B\big) \cdot \big(T_A \otimes T_B\big). \end{aligned}$$

$D_A \otimes D_B$ is a diagonal matrix

$$D_A \otimes D_B = \begin{bmatrix} \|\widehat{\boldsymbol{a}}_1\| \cdot D_B & & 0 \\ & \cdots & \\ 0 & & \|\widehat{\boldsymbol{a}}_{n_A}\| \cdot D_B \end{bmatrix} \qquad D_B = \begin{bmatrix} \|\widehat{\boldsymbol{b}}_1\| & & 0 \\ & \cdots & \\ 0 & & \|\widehat{\boldsymbol{b}}_{n_B}\| \end{bmatrix}$$

where $n_A := \dim \mathcal{L}(A)$ and $n_B := \dim \mathcal{L}(B)$. Applying Proposition 6 yields that $O_A \otimes O_B$ is an orthogonal matrix, too. Finally, it is straightforward to verify that $T_A \otimes T_B$ is an upper triangle matrix with diagonal elements equal to 1. Hence, the diagonal coefficients of $D_A \otimes D_B$ are the heights of the base C. Using the lower bound for the entries of D_A, D_B we get the desired lower bound for the heights of the base C. This also proves Proposition 3 as

$$\det C = \det\big(D_A \otimes D_B\big) \cdot \underbrace{\det\big(O_A \otimes O_B\big)}_{=\pm 1} \cdot \underbrace{\det\big(T_A \otimes T_B\big)}_{=1}$$

and $\det(\mathcal{L}(A) \otimes \mathcal{L}(B)) = |\det C|$.

C Choosing a Random Lattice

We used the term "random lattice" in a sloppy way not precisely defining it. The main reason is that there is no canonical uniform distribution for lattices. One approach is to set up the distribution in terms of lattice bases in a given unique normal form like the *Hermite normal form* [Co93,DKT87]. A matrix $B = [b_{ij}]$ with full row rank is in Hermite normal form, if

1. B is an lower triangular matrix[5] and
2. $0 \le b_{ij} < b_{ii}$ for $j < i$.

Every lattice has a unique lattice base in Hermite normal form where the product of the diagonal elements equals the lattice determinant. To choose a random lattice base B, first select random diagonal coefficients from a given interval $b_{ii} \in_{\mathrm{R}} [1, r]$ and afterwards select the other non-zero coefficients, i.e. $b_{j,i} \in_{\mathrm{R}} [0, b_{ii})$. This method can be used for generating the underlying lattices for the HKZ-Tensor-Trapdoor introduced in Section 5.

But for the Tensor-Hiding-Trapdoor we require some side information about the first successive minimum, e.g. $\lambda_1(L) = \Theta(\sqrt{n})$ for some small constants. We

[5] Some authors define it in terms of an upper triangular matrix.

relax our definition of "random lattices" because to the best of our knowledge given a base in normal form one cannot derive a bound for the first successive minimum beside Minkowski's upper bound. A well understood class of integer lattices is given by modular homogeneous linear equations [FK89]:

$$L_{\boldsymbol{a},m} := \left\{ \boldsymbol{x} \in \mathbb{Z}^n \;\middle|\; \sum_{i=1}^{n} x_i a_i \equiv 0 \pmod{m} \right\}.$$

Paz and Schnorr [PS87] have shown that any integer lattice can be "approximated" by such a lattice. $L_{\boldsymbol{a},m}$ is a n-dimensional lattice with determinant equal to $m/\gcd(a_1, \ldots, a_n, m)$. We restrict ourselves to prime moduli turning $\mathbb{Z}/m\mathbb{Z}$ into a finite field. Fix a prime module p. What is the probability for $\boldsymbol{a} \in_{\mathrm{R}} [1, p)^n$ that $\lambda_1(L_{\boldsymbol{a},p}) \leq k$ for a given threshold k? Given a non-zero $\boldsymbol{x} \in S_n(k) \cap \mathbb{Z}^n$ (where $S_n(k)$ denotes the sphere around the origin with radius k), let i with $x_i \neq 0$. Then $\boldsymbol{x} \in L_{\boldsymbol{a},p}$ iff

$$x_i a_i \equiv -\sum_{j \neq i} a_j x_j \pmod{p}.$$

As p is a prime, given $\boldsymbol{x}$ and a_j for $j \neq i$, there exists exactly one solution a_i in $[0, p)$. This yields:

$$\Pr_{\boldsymbol{a} \in_{\mathrm{R}} [1,p)^n} [\lambda_1(L_{\boldsymbol{a},p}) \leq k] < \frac{|S_n(k) \cap \mathbb{Z}^n|}{p}$$

It is by no means trivial to derive a (sharp) bound for the number of integer points in a sphere [MO90]. But for our purpose it is sufficient to approximate the number by the volume of the sphere:

$$\mathrm{vol}_n\left(S_n(k)\right) = k^n \cdot \mathrm{vol}_n\left(S_n(1)\right) = \frac{k^n \cdot \pi^{\frac{n}{2}}}{\Gamma\left(1 + \frac{n}{2}\right)}$$

If $k = c_0\sqrt{n}$ for a small constant c_0, then for sufficient large n

$$\Pr_{\boldsymbol{a} \in_{\mathrm{R}} [1,p)^n} \left[\lambda_1(L_{\boldsymbol{a},p}) \leq c_0\sqrt{n}\,\right] \leq \frac{2^{c_1 n}}{p}$$

where $c_1 > 0$ is a moderate constant, too. Now, choose a random $(c_1 + 1)n$ bit prime p and $\boldsymbol{a} \in_{\mathrm{R}} [1, p)^n$, then with probability $1 - 2^n$

$$c_0\sqrt{n} < \lambda_1(L_{\boldsymbol{a},p}) < \sqrt{n} \cdot 2^{c_1 + 1 + \frac{1}{n}}$$

where the upper bound is derived using $\det L_{\boldsymbol{a},p} = p < 2^{(c_1+1)n+1}$ and Minkowski's Proposition 1.