Diagonally dominant matrices for cryptography

No Author Given

No Institute Given

Abstract. Diagonally dominant lattices have already been used in cryptography, notably in the GGH and DRS schemes. This paper further studies the possibility of using diagonally dominant matrices in the context of lattice-based cryptography. To this end we study geometrical and algorithmic properties of lattices generated by such matrices. We prove novel bounds for the first minimum and the covering radius with respect to *the max norm*. Using these new results, we propose an decryption failure free encryption scheme using diagonally dominant matrices. We then propose solutions to patch the DRS signature scheme, in particular using parameters leading to the use of sparse matrices.

Keywords: Diagonally dominance · Euclidean lattices · Algorithmic · Statistical attacks.

1 Introduction

1.1 Context and motivation

Diagonally dominant matrices. Diagonally dominant matrices have been an interesting object of study for over a century, starting at least from the Lévy-Desplanques theorem (1881) [1], with several links to general matrix theory with research spanning up to today [29,12,52]. Numerous applications of diagonal dominance can be found in various fields such as numerical linear algebra [36], Markov chains, graphs Laplacians, perturbation theory[2]. On the other hand, lattices generated by diagonally dominant matrices fitting the Lévy-Desplanques theorem was not investigated. Such lattices seemed to have found some application in cryptography on few specific instances [49,55] where in both papers the focus was more in the matrix generation than a study of the resulting lattice. On the other hand, when strict dominance is not required (i.e not fitting the Lévy-Desplanques theorem), "large diagonals" saw some uses in cryptography [30,40,50] as well as in modular arithmetic [6].

Euclidean lattices. The study of computational problems on lattices in general is also an old and very studied topic [44,14,5]. Classical problems such as computing a shortest vector – named the Shortest Vector Problem (SVP) – and computing the closest lattice vector from a target vector – the Closest Vector Problem

[1] A history of this theorem through the ages can be seen in [57]

[2] [19] lists some applications.

(CVP) – can be proven to be NP-hard in the general case [1,39]. As a matter of fact, relaxed version of these problems stay hard. Notably, even if we authorise exponential preprocessing computations, the CVP is also NP-hard for small approximation factors [3]. The hardness of these problems over Euclidean lattices motivated cryptographers to consider them as building blocks for cryptographic schemes [31,51], which led to extensive study of Euclidean lattices in the past decades.

Lattice-based cryptography. The first example of schemes using Euclidean lattices were using generic lattices and use a trapdoor one-way function whose hardness to invert is based on the CVP. One can cite the Goldreich-Goldwasser-Halevi (GGH) scheme [31] or constructions using the plain Learning With Errors (LWE) problem such as FRODO [11]. Note that their security can also be linked to the hardness of the SVP. For efficiency reasons one tends to consider *algebraic lattices*, meaning lattices which can be described by means of polynomial rings. Some of the noticeable constructions are NTRU [32] or the schemes based on the Ring Learning With Errors (RING-LWE) or the Module Learning With Errors (MODULE-LWE) problems. Their security can be linked to the SVP on the restricted classes of *ideal lattices* – also called the Ideal Shortest Vector Problem (IDEAL-SVP) – or *module lattices* – also called the Module Shortest Vector Problem (MODULE-SVP). One may wonder whether the additional algebraic structure can be used to solve the SVP more efficiently. Thus, the study of the IDEAL-SVP has gathered sustained attention in the past few years. First it was shown that the intermediate problem of recovering short generators of principal ideals can be solved in quantum polynomial time over cyclotomic fields [15] and even classical polynomial time over multiquadratic [8] or multicubic fields [35]. Then Cramer, Ducas and Wesolowski extended the analysis of [15] to the IDEAL-SVP and showed that one could obtain a subexponential approximation factor in quantum polynomial time [16]. With a slightly different approach, this result can be generalized to all number fields provided an exponential pre-processing phase [48], which might be an artifact of the proof if we refer to experimental results obtained in [9,10]. Thus the IDEAL-SVP seems to be strictly weaker than the SVP. Even though the RING-LWE or MODULE-LWE problems are harder than the IDEAL-SVP, there is no guarantee that algebraic attacks mentioned previously cannot be used to tackle them.

Thus, studying other types of trapdoors or constructions is still an interesting and important research direction, recently explored in [26] or [22,24] for example.

Digital signatures with lattices. In order to build digital signatures schemes with lattices, one can follow the *hash-then-sign* paradigm. In this setting, the hash of the message $H(m)$ is a random vector of the space and a valid signature is then a lattice vector close to $H(m)$. The security of the scheme is guaranteed as soon as solving the CVP is hard. The original GGH and NTRU signature schemes were originally following a naive version of this paradigm, using the so-called Babai round-off algorithm to produce the signature. However Nguyen and Regev successfully used the observation that the difference between the message and a valid signature lie within the fundamental parallelepiped of the secret basis to recover the latter [45]. Ducas and Nguyen showed that this statistical attack could be extended to more complex structures than bases which allowed them to break potential counter-measures in practice [21]. The same kind of attack [37] has recently been applied to break the PEREGRINE signature scheme [53].

In order to prevent the attack, Plantard, Win and Susilo [50] described how to produce a hash-then-sign scheme based on the max norm in the hope that the signatures lie in a space independent of the secret basis. Their work rely on matrices of the form $\mathbf{B} = \mathbf{D} + \mathbf{N}$ where $\mathbf{D}$ and $\mathbf{N}$ are such that the spectral radius $\rho(\mathbf{D}^{-1} \cdot \mathbf{N}) < 1$. Then this work has been adapted for DRS, a candidate of the first round of the NIST call for standardization [49], relying on the fact that the matrices used as lattice bases are diagonally dominant. This allows the γ-Guaranteed Distance Decoding (GDD_γ) to be solved with an algorithm adapted from [50]. This scheme has known a learning attack by Ducas and Yu [23]. One has to note that this attack differs from the previous ones and that it *does not break completely* the second version of the scheme [56]. However, it remains a serious attack with around 30 bits of security loss for the first set of parameters, using 2^{30} signatures only.

1.2 Our Contributions

This work is composed of three parts.

1. In Section 3 we improve our theoretical knowledge of diagonally dominant lattices by giving two new bounds on the key lattice invariants in the context of cryptography *for the max norm*, one for the covering radius and one for the first minimum.
   More precisely, we start by giving a lower bound on the size of the shortest vector in infinity norm. Guessing the size of the shortest vector or even an approximation is known to be NP-hard [18], thus we believe providing a tighter upper bound for any specific family of lattices is an interesting result in itself. Then we give an improved study of the reduction algorithm of [50] for diagonally dominant matrices and prove a stronger reduction capability than previously proven for such lattices [55]. We also prove that our aforementioned algorithms operate at most a polynomial (in the dimension and the size of its entries) amount of vector additions or multiplications by a scalar. Consequently, both results give novel upper and lower bounds on the size of the covering radius for such lattices
2. Secondly, using this new results, we are able to provide a decryption failure free cryptosystem relying on diagonally dominant matrices. It follows a framework close the GGH encryption schemes [31,7]. We discuss formal security and the steps to take towards IND-CCA security, using standard techniques or transformations [28,17]. We also evaluate the practical security of the scheme using common cryptanalytic techniques to assess lattice-based constructions. We show that it is asymptotically secure.

3. Finally, following a tighter cryptanalysis, we explore a modification of the DRS scheme using sparse matrices. We notably look into the impact of Ducas and Yu's statistical attack [23]. Our experiments tend to show that these new parameters greatly mitigate the impact of the leak.

Conclusion. We deem that the asymptotical security of GGH-like schemed using diagonally dominant matrices can be achieved, however our work tends to show that the practical security at a level comparable to other schemes like SQUIRRELS or HAWK [24] is difficult to achieve for suitable dimensions. Thus, we deem that trying to achieve such a goal is still an interesting and challenging research direction. Another option that we plan to explore is to use the good decoding properties of such matrices in other framework such as the Lattice Isomorphism Problem [22].

2 Background

We assume the readers know what is the set of integers $\mathbb{Z}$, the set of integral matrices with n rows and m columns $M_{n,m}(\mathbb{Z})$, the determinant, norms and other basics of linear algebra. Given a matrix $\mathbf{B}$, we will denote by $\mathbf{B}_i$ its *ith* row vector. We refer readers to [41,42] for a more complete background of lattice theory.

Definition 1 (Lattice).
We define an integral lattice $\mathcal{L}$ as a subgroup of $\mathbb{Z}^n$. A basis $\mathbf{B}$ of an integral lattice $\mathcal{L}$ is a basis of $\mathcal{L}$ as a $\mathbb{Z}$-module, and denote by $\mathcal{L}(\mathbf{B})$ the lattice generated by the rows of a basis $\mathbf{B}$. We write the volume (or determinant) of the lattice and compute it as $\det(\mathcal{L}) = \sqrt{\det(\mathbf{B} \cdot \mathbf{B}^T)}$.

While an integral lattice can potentially have an infinity of basis, a lattice only admits an unique basis in Hermite Normal Form (HNF).

Definition 2 (HNF).
Let $\mathcal{L}$ be a full-rank integral lattice of dimension n and $\mathbf{H} \in M_{n,n}(\mathbb{Z})$ a basis of $\mathcal{L}$. Then $\mathbf{H}$ is said to be in HNF *if, and only if,*

$$\forall 1 \leq i,j \leq d,\ \mathbf{H}_{i,j} \begin{cases} = 0 & if\ i > j \\ \geq 0 & if\ i \leq j \\ < \mathbf{H}_{j,j} & if\ i < j \end{cases}$$

In this paper we only consider full-rank integral lattices.

Lattices have some important invariant with strong computational property.

Definition 3 (Minima of a lattice). *We denote by $\lambda_k^{(l)}(\mathcal{L})$ the smallest value r such that a ball centered in zero and of radius r in norm l contains k linearly independent vectors of $\mathcal{L}$.*

Definition 4 (Covering radius). *Given a lattice $\mathcal{L}$, we define its covering radius $\mu^{(l)}(\mathcal{L})$ as the smallest value such that for any $\mathbf{x} \in \mathbb{R}^n$, there exists $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{x} - \mathbf{v}\|_l < \mu^{(l)}(\mathcal{L})$.*

There is some relation between all those invariants. For example, for any lattice $\frac{1}{2}\lambda_1^{(2)}(\mathcal{L}) \leq \mu^{(2)}(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n^{(2)}(\mathcal{L})$ (See [41]).

While many computational problems on lattices exist, we define only the lattice problems useful for the comprehension of the paper.

Definition 5 (Approximate Shortest Vector Problem (SVP$_\gamma$)). *Given a basis of a lattice $\mathcal{L}$ of dimension n and an* approximation factor $\gamma \in \mathbb{R}_+$*, find $\mathbf{v} \in \mathcal{L} \setminus \{0\}$ such that $\|v\ Vert \leqslant \gamma \cdot \lambda_1(\mathcal{L})$.*

Definition 6 (Approximate Closest Vector Problem (CVP$_\gamma$)). *Given a basis of a lattice $\mathcal{L}$ of dimension n, a target vector $\mathbf{t} \in \mathbb{R}^n$ and an approximation factor $\gamma \in \mathbb{R}_+$, find $\mathbf{v} \in \mathcal{L}$ such that $\forall \mathbf{w} \in \mathcal{L}, \|\mathbf{t} - \mathbf{v}\| \leq \gamma \cdot \|\mathbf{t} - \mathbf{w}\|$.*

The first minimum $\lambda_1^{(l)}(\mathcal{L})$ and the covering radius $\mu^{(l)}(\mathcal{L})$ offers some natural bounds which transform the generic problem CVP in some useful variant, especially for cryptographic applications.

Definition 7 (GDD$_\gamma$). *Given a lattice $\mathcal{L}$, and a bound $\gamma \geq 1$, for any target $\mathbf{t} \in \mathbb{R}^n$ find a lattice vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{t} - \mathbf{v}\| < \gamma \cdot \mu^{(l)}(\mathcal{L})$.*

There exists another variant of CVP; if the first variant, GDD$_\gamma$, is key for lattice based signature scheme, the second variant, Bounded Distance Decoding (BDD), us key for lattice based encryption scheme.

Definition 8 (BDD). *Given a lattice $\mathcal{L}$, and a bound $\alpha \leq 1$, for any target $\mathbf{t} \in \mathbb{R}^n$ such there exist a vector $\mathbf{v} \in \mathcal{L}$ with $\|\mathbf{t} - \mathbf{v}\| < \alpha \cdot \lambda_1^{(l)}(\mathcal{L})$, find $\mathbf{v}$.*

Those problems are usually tackled with the combination of a "good" basis, e.g. LLL-reduced [34] or BKZ-reduced [13], together with an appropriate algorithm such as Babai's round-off or nearest plane algorithms [5]. For example, that is the approach proposed by Klein [33] for solving BDD for some α.

Remark 1. Note that a CVP$_\gamma$ algorithm can be used as a GDD$_\gamma$ solver as long as the approximation factor γ ensures that any target has a solution. Remark also that solving the GDD$_\gamma$ is equivalent to computing a *short* coset representative of $\mathbf{t} \bmod \mathcal{L}$. We will often consider algorithms solving this "short coset representative" problem, that we will call *reduction algorithms* and write `Reduce` for a generic algorithm. In this context the approximation factor γ of Definition 7 will be called the *reduction radius*.

In this paper, we consider a specific family of "good" lattice bases, allowing us to tackle the above problems more easily. Thus, we can use them as secret trapdoors for cryptographic constructions.

Definition 9 (Diagonally Dominant Matrix). *Let a matrix* $\mathbf{B} \in M_n(\mathbb{Z})$, *we write* $\delta_i(\mathbf{B})$,

$$\delta_i(\mathbf{B}) = \mathbf{B}_{i,i} - \sum_{\substack{j=1 \\ i \neq j}}^{n} |\mathbf{B}_{i,j}|$$

and we will call $\mathbf{B}$ diagonally dominant *if, and only if,*

$$\forall i \in [\![1, n]\!], \quad \delta_i(\mathbf{B}) > 0.$$

Furthermore, we will call dominance level *the quantity* $\Delta(\mathbf{B}) \stackrel{\text{def}}{=} \min \delta_i(\mathbf{B})$.

It follows from the Lévy-Desplanques theorem that a diagonally dominant matrix is always full-rank.

For clarity reasons, we will mainly consider diagonally dominant matrices of the form $\mathbf{B} = D \cdot \mathrm{Id}_n + \mathbf{N}$ for some fixed $D \in \mathbb{Z}$ and such that for any $i \in [\![1, n]\!]$, $\mathbf{N}_{i,i} = 0$ [3]. Then, we will call *noise level* the value $\nu(\mathbf{B}) \stackrel{\text{def}}{=} \max_{i \in [\![1,n]\!]} \|\mathbf{N}_i\|_1$.

3 Results on fundamental values for diagonally dominant lattices

In this section we analyze diagonally dominant lattices with respect to *the max norm*. We improve our knowledge on both the covering radius and the first minimum which are cryptographically relevant lattice invariants. We present those results in Theorem 1 and Theorem 2. Moreover we show that the bound for the covering radius for matrices with negative noise $\mathbf{N}$ can be lowered, but we push back this result in Appendix A for clarity purposes.

3.1 Tighter bound on Diagonally Dominant Lattice Covering Radius

The results proven in this section will prove the following theorem.

Theorem 1. *Consider* $\mathbf{B} \in M_n(\mathbb{Z})$ *a diagonally dominant matrix and* $\mathcal{L} = \mathcal{L}(\mathbf{B})$. *There is an algorithm* `PSW` *(Alg. 1) such that for any vector* $\mathbf{v} \in \mathbb{R}^n$, *it returns in polynomial time a vector* $\mathbf{w}$ *respecting*

$$\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}, \quad \|\mathbf{w}\|_\infty \leqslant D - \frac{\Delta(\mathbf{B})}{2}$$

i.e.

$$\mu^{(\infty)}(\mathcal{L}) \leqslant D - \frac{\Delta(\mathbf{B})}{2}.$$

[3] Note however that our results and their proofs can be adapted to the case where $\mathbf{B} = \mathbf{D} + \mathbf{N}$ with $\mathbf{D}$ a general diagonal matrix and $\mathbf{N}$ has non-zero diagonal coefficients.

The proof of this theorem is done by proving an upper bound on the convergence radius of a reduction algorithm which we will prove to terminate within a polynomial number of arithmetic operations.

The `PSW` reduction algorithm was first introduced in [50] and is a known approximation of Babaï's Round-off algorithm [5] in the case of matrices of the form $\mathbf{D}-\mathbf{N}$ where $\mathbf{N}\cdot\mathbf{D}^{-1}$ have a spectral radius lower than 1. It was then used a second time in cryptography [49] within the DRS scheme. The algorithm was proven to finish for with $\|\mathbf{w}\|_\infty < D$ in [49], but did not take into account the leeway $\Delta(\mathbf{B})$ A slight modification of the reduction proof given in [55] gives us a tighter bound by changing the loop condition in line 2 of the algorithm to a comparison with a value $R_i \geqslant D - \delta(\mathbf{B}, i)/2$ for every index i. This gives us the modified version, described in Algorithm 1.

Algorithm 1 `PSW` reduction

Require: $\mathbf{v} \in \mathbb{R}^n, \mathbf{B}$ a diagonally dominant matrix, a bound vector $R \in \mathbb{N}^n$.
Ensure: $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(B)$ and $\forall i \in [\![1, n]\!], \mathbf{w}_i < R_i$.

1: $\mathbf{w} \leftarrow \mathbf{v}$
2: **while** $\bigvee_{j=1}^{n}(|\mathbf{w}_j| > R_j)$ **do**
3:     $i \leftarrow$ any index such that $|\mathbf{w}_i| > R_i$
4:     **if** $|\mathbf{w}_i| \geqslant D$ **then**
5:         $q \leftarrow \mathrm{sign}(\mathbf{w}_i)\cdot \lfloor |\mathbf{w}_i| / D \rfloor$
6:     **else**
7:         $q \leftarrow \mathrm{sign}(\mathbf{w}_i)$
8:     **end if**
9:     $\mathbf{w} \leftarrow \mathbf{w} - q \cdot \mathbf{B}_i$ {Reduce $|\mathbf{w}_i|$}
10: **end while**
11: **return** $\mathbf{w}$

Correctness. The following lemma states that for a given R, the algorithm terminates given that values R_i are above a certain bound which varies for each index.

Lemma 1 (Tighther bound in `PSW`-reduction algorithm). *For input $\mathbf{v} \in \mathbb{Z}^n$, a diagonally dominant matrix $\mathbf{B}$ and $R \in \mathbb{R}_+^n$ such that $\forall i \in [\![1, n]\!], R_i \geqslant D - \delta_i(\mathbf{B})/2$, the `PSW` reduction (alg. 1) terminates and outputs $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(\mathbf{B})$ where $\forall i,\ |\mathbf{w}_i| \leq R_i$.*

Proof. Let $S(\mathbf{v}, R) \stackrel{\text{def}}{=} \{i \in [\![1, n]\!] \mid |\mathbf{v}_i| > R_i\}$ and f be the function defined on $\mathbb{Z}^n \times [\![1, n]\!]$ by $f : (\mathbf{w}, i) \mapsto \mathbf{w} - \mathrm{sign}(\mathbf{w}_i) \cdot \left\lfloor \frac{\mathbf{w}_i}{D} \right\rfloor \cdot \mathbf{B}_i$. In order to show that Algorithm 1 ends and outputs a correct vector, we will prove the following:

$$\bigvee_{j=1}^{n}(|\mathbf{w}_j| > R_j) \implies \forall i \in S(\mathbf{w}, R), \|f(\mathbf{w}, i)\|_1 < \|\mathbf{w}\|_1. \tag{1}$$

First remark that if the left side of (1) is verified, then f modifies $\mathbf{w}$. Now let us show that (1) is true. First assume that there exists $i \in S(\mathbf{w}, R)$ such that $|\mathbf{w}_i| > D$. Then $f(\mathbf{w}, i)_i$ has the same sign than $\mathbf{w}_i$, therefore $|f(\mathbf{w}, i)_i| = |\mathbf{w}_i| - \lfloor |\mathbf{w}_i| / D \rfloor \cdot D$. Moreover we have

$$\forall j \in [\![1, n]\!] \setminus \{i\}, |\mathbf{w}_j| \leqslant |\mathbf{w}_j| + \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot |\mathbf{B}_{i,j}|,$$

which gives

$$\|f(\mathbf{w}, i)\|_1 \leqslant |f(\mathbf{w}, i)_i| + \sum_{\substack{j=1 \\ j \neq i}}^{n} |f(\mathbf{w}, i)_j| \leqslant |\mathbf{w}_i| - \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot D + \sum_{\substack{j=1 \\ j \neq i}}^{n} |\mathbf{w}_j| + \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot |\mathbf{B}_{i,j}|.$$

This leads to

$$\|f(\mathbf{w}, i)\|_1 \leqslant \|\mathbf{w}\|_1 + \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot \delta_i(\mathbf{B}) \leqslant \|\mathbf{w}\|_1 - \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot \delta_i(\mathbf{B}) < \|\mathbf{w}\|_1.$$

Now consider $i \in S(\mathbf{w}, R)$ such that $|\mathbf{w}_i| < D$. Then the signs of $\mathbf{w}_i$ and $f(\mathbf{w}, i)_i$ are different. Moreover if we write $|\mathbf{w}_i| = R_i + t$ with $t \in [\![1, D - R_i[\![$, we obtain $|f(\mathbf{w}, i)_i| = |R_i - D + t| = D - R_i - t$. Therefore we have

$$|f(\mathbf{w}, i)_i| = |\mathbf{w}_i| - 2(R_i + t) + D.$$

Following the same reasoning as before to bound $\|f(\mathbf{w}, i)\|_1$, we have

$$\|f(\mathbf{w}, i)\|_1 \leqslant \|\mathbf{w}\|_1 - 2(R_i + t) + D + D - \delta_i(\mathbf{B})$$

and noting that $R_i \geqslant D - \delta_i(\mathbf{B})/2$ we obtain

$$\|f(\mathbf{w}, i)\|_1 \leqslant \|\mathbf{w}\|_1 - 2(R_i + t) + 2R_i < \|\mathbf{w}\|_1.$$

□

Algorithm 1 uses a linear memory and does not need to store much more than the size of the target and the matrix. This is an advantage compared to Babaï's nearest plane algorithm which needs the GSO or Babaï's rounding-off algorithm which requires a matrix inverse. Moreover all computations can be carried out with simple integral arithmetic.

Worst-case complexity. The average-case time-complexity of Algorithm 1 was briefly experimented in [50], however a proper worst-case analysis was not provided and does not seem to have been done in the literature.

Lemma 2. *Let $\mathbf{B} \in M_n(\mathbb{Z})$ be a diagonally dominant matrix and $\mathbf{v} \in \mathbb{Z}^n$, and denote by b the value $\frac{2nD}{2nD - \Delta(\mathbf{B}))}$. An upper bound on the complexity of vector operations done by Algorithm 1 is in*

$$O\left(\log_b\left(\frac{\|\mathbf{v}\|_1}{nD}\right) + \frac{nD}{2}\right).$$

Proof. Let us consider the reduction of $\|\mathbf{w}\|_1$ to count the number of reduction steps, using the results and the reasoning of Lemma 1.

First assume $\|\mathbf{w}\|_1 > nD$ which guarantees $\|\mathbf{w}\|_\infty > D$. Thus the coefficient q is greater 1. Denote by $\mathbf{w}'$ the value of the vector after the update in Algorithm 1. Then $\|\mathbf{w}\|_1$ is updated as

$$\|\mathbf{w}'\|_1 \leqslant \|\mathbf{w}\|_1 - q \cdot \Delta(\mathbf{B}).$$

From $\|\mathbf{w}\|_\infty \leqslant \|\mathbf{w}\|_1 \leqslant n\|\mathbf{w}\|_\infty$ we obtain $q \geqslant \frac{\|\mathbf{w}\|_1}{2nD}$. Thus we get

$$\|\mathbf{w}'\|_1 \leq \|\mathbf{w}\|_1 - \frac{\|\mathbf{w}\|_1}{2nD} \cdot \Delta(\mathbf{B}) = \|\mathbf{w}\|_1 \cdot \left(\frac{2nD - \Delta(\mathbf{B})}{2nD}\right).$$

If we use this inequality and we write k the number of steps necessary to reach the condition $\|w\|_1 \leqslant nD$, i.e to reach the second case, using the worst assumptions we obtain:

$$\|\mathbf{w}\|_1 = \left(\frac{2nD - \Delta(\mathbf{B})}{2nD}\right)^k \cdot \|\mathbf{v}\|_1 \leqslant nD.$$

This gives a $O\left(\log_b\left(\frac{\|\mathbf{v}\|_1}{2nD}\right)\right)$ number of vectors operations to reach $\|\mathbf{w}\|_1 \leqslant nD$.

We can now focus on the case $\|\mathbf{w}\|_1 \leqslant nD$. Note that $\|\mathbf{w}\|_1 \leqslant nD$ still do not give us much information about $\|\mathbf{w}\|_\infty$, so we continue our analysis using $\|\mathbf{w}\|_1$. We proceed by counting the least untactful possible reduction of $\|\mathbf{w}\|_1 \leqslant nD$ per step until $\|\mathbf{w}\|_1 = 0$: each step reduces $\|\mathbf{w}\|_1$ of at least $2t$ (2 with $t = 1$). Therefore, we upper-bound the amount of loop iterations left by $\frac{\|\mathbf{w}\|_1}{2} \leqslant \frac{nD}{2}$. □

By approximating $\log(b) = -\log(1 - \Delta/2nD) \approx \Delta(\mathbf{B})/2nD$ and setting $\|\mathbf{v}\|_1 = nD^n$ (i.e each coefficient to an approximate of the determinant), we can obtain the simpler formula ignoring constants:

$$O\left(n^2 D \frac{\log(D)}{\Delta(\mathbf{B})}\right)$$

In addition, if we set $D = n$ and $\Delta(\mathbf{B}) = 1$ as in the different versions of the DRS scheme [49,55,56], we obtain $O(n^3 \log n)$.

Remark 2. This complexity bound obtained in Lemma 2 is not tight and does not reflect at all the significantly faster experimental results reported in [50,55,49], which is understandable: the probability to trigger a *single* least-impactful iteration is $2^{-(n-1)}$, i.e as probable as solving a $\{0,1\}$-knapsack problem with $n-1$ entries randomly. However, our result still proves polynomial operation complexity and constant memory (besides input memory) as far as vector operations (i.e. fixed dimension) are concerned.

3.2 Result on Diagonally Dominant Lattice First Minimum

The importance of $\Delta(\mathbf{B})$ for the quality of the lattice have been exposed in the previous section. In this section, we present a second result linking once again

$\Delta(\mathbf{B})$ with an invariant of the lattice. However, this time we are able to bound the first minima of the lattice. This is the first result in this direction which alleviate the complexity of using diagonally dominant matrix for encryption, especially if one wants to avoid any probability of decryption failure.

Theorem 2. *Let $\mathbf{B} \in M_n(\mathbb{Z})$ be a diagonally dominant matrix with diagonal D. Then $\lambda_1^{(\infty)}(\mathcal{L}(\mathbf{B})) \geqslant \Delta(\mathbf{B})$.*

Proof. Consider $l \in \mathbb{Z}^n$ and write $\mathbf{v} = l \cdot \mathbf{B}$. Then write $l' = (|l_i|)_{i \in [\![1,n]\!]}$. There exists $\mathbf{B}' \in M_n(\mathbb{Z})$ a matrix such that $|B'_{i,j}| = |B_{i,j}|$ for any pair $(i,j) \in [\![1,n]\!]^2$, and for all $i \in [\![1,n]\!], \mathbf{B}'_{i,i} = D$ and $\mathbf{v}_i = \pm(l' \cdot \mathbf{B}')_i$. Thus $\mathbf{B}'$ is a diagonally dominant matrix such that $\delta_i(\mathbf{B}') = \delta_i(\mathbf{B})$ for all $i \in [\![1,n]\!]$. Now let us show that $\|\mathbf{v}\|_\infty \geqslant \Delta(\mathbf{B})$. We will first bound the taxicab norm then use the classic norm inequality

$$\|\mathbf{v}\|_\infty \leqslant \|\mathbf{v}\|_1 \leqslant n\|\mathbf{v}\|_\infty. \tag{2}$$

First remark that we have the following:

$$\|\mathbf{v}\|_1 = \sum_{j=1}^{n} |(l' \cdot \mathbf{B}')_j| \geqslant \left| \sum_{j=1}^{n} \sum_{i=1}^{n} l'_i \cdot \mathbf{B}'_{i,j} \right| = \left| \sum_{i=1}^{n} l'_i \sum_{j=1}^{n} \mathbf{B}'_{i,j} \right|.$$

Moreover for any $i \in [\![1,n]\!]$, $l'_i \geqslant 0$ and $\sum_{j=1}^{n} \mathbf{B}'_{i,j} \geqslant \delta_i(B) > 0$, so we have

$$\left| \sum_{i=1}^{n} l'_i \sum_{j=1}^{n} \mathbf{B}'_{i,j} \right| = \sum_{i=1}^{n} l'_i \sum_{j=1}^{n} \mathbf{B}'_{i,j} \geqslant \sum_{i=1}^{n} l'_i \cdot \delta_i(\mathbf{B}).$$

Therefore, if $k = |\{i \in [\![1,n]\!] \mid l_i \neq 0\}|$ we obtain $\|\mathbf{v}\|_1 \geqslant k \cdot \Delta(\mathbf{B})$.
If $k = n$ then Equation (2) gives

$$\|\mathbf{v}\|_\infty \geqslant \Delta(\mathbf{B}).$$

Now consider the case with $k < n$. Without any loss of generality, assume $\forall i \in [\![1,k]\!], l_i \neq 0$. Denote by l'' the tuple $(l'_1, \ldots, l'_k)$ and $\mathbf{B}''$ the top left $k \times k$ submatrix of $\mathbf{B}'$. Then $\mathbf{B}''$ is diagonally dominant and $\forall i \in [\![1,k]\!], \delta_i(\mathbf{B}'') \geqslant \delta_i(\mathbf{B}') = \delta_i(\mathbf{B})$. We have

$$\forall i \in [\![1,k]\!], (l \cdot \mathbf{B})_i = (l' \cdot \mathbf{B}')_i = (l'' \cdot \mathbf{B}'')_i.$$

Then, since $|\{i \in [\![1,k]\!] \mid l''_i \neq 0\}| = k$, we can apply the previous result to l'' and $\mathbf{B}''$, therefore $\|l'' \cdot \mathbf{B}''\|_\infty \geqslant \Delta(\mathbf{B}'')$ and $\exists i_0 \in [\![1,k]\!], |(l'' \cdot \mathbf{B}'')_{i_0}| = \|l'' \cdot \mathbf{B}''\|_\infty$. Finally we get

$$|(l \cdot \mathbf{B})_{i_0}| = |(l' \cdot \mathbf{B}')_{i_0}| = |(l'' \cdot \mathbf{B}'')_{i_0}| \geqslant \Delta(\mathbf{B}'') \geqslant \Delta(\mathbf{B}') = \Delta(\mathbf{B}).$$

□

4 Diagonally Dominant Matrix Encryption

In this section we will describe an encryption scheme using diagonally dominant matrices, the we call DRE as a callback to DRS. First we describe in Section 4.1 the general framework of our construction based on a GDD_γ solver. We provide conditions on the matrices used as private keys to ensure the correctness of the scheme within this framework in Section 4.2. To this end we use the results on $\lambda_1^{(\infty)}$ and $\mu^{(\infty)}$ proven in Section 3 and summed-up in Theorem 1. Then we give an instantiation of this general framework in Section 4.3 and discuss security in Sections 4.4 and 4.5.

4.1 General framework

Let us now describe the framework for the encryption scheme we are considering. As mentioned previously, it is based on the max norm l_∞. We fix as parameters $(D, n, M) \in \mathbb{N}^2$. Let us denote $\mathcal{L}$ the lattice generated by a diagonally dominant matrix $\mathbf{B} = D \cdot \mathrm{Id}_n + \mathbf{N}$. Let R be the radius in which we can find for any $\mathbf{c} \in \mathbb{Z}^n$ a vector $\mathbf{m} \equiv \mathbf{c} \in \mathcal{L}$ s.t. $\|\mathbf{m}\|_\infty < R$. Algorithms 1 and 3 offers us parametrisable radii R directly from a parametrisable $\mathbf{B}$. Evidently, $\mathbf{B}$ is kept as a secret trapdoor as it allows for decryption. Let M be the upper bound of the max norm of the vector messages we wish to recover, such that if the vectors associated to the valid messages belong to a set $\mathcal{M}$, then $\mathcal{M} \subseteq [-M, M]^n$. Here, we consider that each message is associated to a vector $\mathbf{m} \in \mathbb{Z}^n$ we wish to recover, and that the encryption of $\mathbf{m}$ is associated to a ciphertext vector $\mathbf{c} = \mathbf{m} + \mathbf{v}$ where $\mathbf{v} \in \mathcal{L}(\mathbf{B})$. In summary we consider the following framework:

- The secret key $\mathbf{S}_K = \mathbf{B} \in M_n(\mathbb{Z})$ is a diagonally dominant matrix with diagonal coefficient D, and the public key $\mathbf{P}_K$ is $\mathbf{H} = \mathrm{HNF}(\mathbf{B})$.
- The message space is $\mathcal{M} \subseteq [\![-M, M]\!]^n$.
- The encryption function will be $\texttt{Encrypt}(\mathbf{m}, \mathbf{P}_K) = s \cdot \mathbf{H} + \mathbf{m}$, with $s \in \mathbb{Z}^n$.
- The decryption function will be $\texttt{Decrypt}(\mathbf{c}, \mathbf{S}_K) = \texttt{Reduce}(\mathbf{c}, \mathbf{B})$, where `Reduce` is a GDD_γ solver. Its convergence radius will be denoted by R.

4.2 Guaranteeing decryption of valid messages (i.e. correctness)

In order to obtain a *correct* scheme we need to determine parameters ensuring the correctness of the decryption. The first condition that they need to satisfy is $M \leqslant R$ so that $\texttt{Reduce}(\mathbf{c}, \mathbf{B})$ is indeed a valid message. Then one needs to ensure unicity, meaning $\texttt{Reduce}(\texttt{Encrypt}(\mathbf{m}, \mathbf{P}_K)) = \mathbf{m}$. This is satisfied as soon as

$$R + M \leqslant \lambda_1^{(\infty)}(\mathcal{L}). \tag{3}$$

In particular for diagonally dominant matrices, we can use Algorithm 1 for `Reduce` and Theorem 1 ensures that Equation (3) can be simply satisfied for $\mathbf{B}$ such that

$$\Delta(\mathbf{B}) > \frac{2}{3}(D + M), \tag{4}$$

which are straightforward to construct.

If we focus on matrices with negative noise only, then we can obtain larger bounds. Indeed, in this case $R = D/2$ so (3) becomes $\lambda_1^\infty(\mathcal{L}) \geqslant D/2 + M$ which gives $\Delta(\mathbf{B}) > D/2 + M$.

Thus, we could use smaller dominance levels for a fixed M or larger message spaces for the same value $\Delta(\mathbf{B})$.

4.3 Instantiation of the encryption scheme

To instantiate our encryption scheme, we first need to fix some public parameters as the diagonal coefficient D and the dimension n. We assume the message space is composed of vectors over $\mathcal{M} = \{-1, 0, 1\}^n$, but we showed earlier that could also be subject to change. From an external point of view, our scheme is is close to knapsack problem, such as the first proposition of Merkle-Hellman [38]. The major difference is within the setup and the decryption, which are details that are hidden to the messages senders.

```
DRE-Setup(D, n)
Δ(B) ← 2(D + 1)/3
H ← 0
while IsNotPerfect(H)
    B ← RDDgen(D, n, Δ(B))
    H ← HNF(B)
P ← PreComp(P_K, S_K)
h ← H[1..n, n]
return (P_K = h, S_K = B, P)
```

```
DRE-Encrypt(m, P_K)
c ← 0
for i ∈ [1, n − 1]
    c ← c − m_i · h_i
c ← (c + m_n) mod h_n
return c
```

```
DRE-Decrypt(c, S_K, P)
Δ(B) ← 2(D + 1)/3
R ← D − Δ(B)/2 · [1, ..., 1]
m ← UsePreComp(c, P)
m ← RSR(m, B, R)
return m
```

Fig. 1: DRE Algorithms

We present above a high-level description of the instantiated encryption scheme in figure 1. The notations for the figure are the following:

- $(D, n, c) \in \mathbb{N}^3$ are respectively the diagonal value, dimension and ciphertext.
- $\mathbf{H}, \mathbf{B} \in \mathbb{Z}^{n\times n}$, and $\mathcal{P}$ is rectangular but has n columns.
- `IsNotPerfect` outputs `TRUE` anytime the input is not a perfect HNF.
- `PreComp` takes the secret key and precompute a set of vectors that are reduction of large integers in the last coefficient.

– `UsePreComp` uses the above to output a vector of smaller integers that is a reduction of an input large integer as a last vector coefficient.

Note that `PreComp` and `UsePreComp` are completely unnecessary from a theoretical perspective, however we use them in our simple proof-of-concept implementation to avoid large waiting times (to verify our concept works in practice), and we thus believe they are worth mentioning. We give below a bit more detailed description of the whole scheme:

Setup. For the secret key, we generate a diagonally dominant matrix with our chosen parameters (D, n). Since the message space is $[\![-1, 1]\!]^n$, following Equation (4), we will fix $\Delta(\mathbf{B}) = \frac{2}{3}(D+1)$. For the public key, we compute the HNF of the secret key, hoping it has perfect form, i.e. $\mathcal{L}(\mathbf{B})$ is a co-cyclic lattice. If not then we discard $\mathbf{B}$ and retry.[4]. The public key is then $\mathbf{H} = \texttt{HNF}(\mathbf{B})$ but since it holds a perfect form, only its unique dense column vector $\mathbf{h}$ needs to be sent. We also use perform precomputations to ease future decryptions, but in theory this is not needed.

Encryption. For the encryption, we just perform a Gaussian Elimination on $\mathbf{m}$ using the matrix $\mathbf{H}$. Because the public key $\mathbf{P}_K = \mathbf{h}$ was enforced to be the last column of a perfect HNF $\mathbf{H}$, the output of `DRE-Encrypt` in figure 1 is the last coefficient of a vector of the form $[0, \ldots, 0, c] = \mathbf{m} + \mathbf{v}$ with $\mathbf{v} \in \mathcal{L}(B)$. Indeed, if one reduces the vector $\mathbf{m}$ with $\mathbf{H}$, as follows

$$\begin{bmatrix} m_1 & \ldots & \ldots & m_{n-1} & m_n \\ \hline 1 & 0 & \ldots & 0 & h_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \ldots & 0 & 1 & h_{n-1} \\ 0 & \ldots & \ldots & 0 & \det(\mathbf{B}) \end{bmatrix},$$

using the first $n-1$ rows of $\mathbf{H}$, the first vector will be transformed into

$$[0, \ldots, 0, m_n - \sum_{i=1}^{n-1} m_i h_i] = \mathbf{m} - m \cdot \mathbf{H} + m_n \cdot [0, \ldots, 0, \det(\mathbf{B})].$$

Note that this approach is very similar to the one chose in the Squirrels scheme [26] recently submitted to the NIST call for proposals for quantum-resistant digital signature algorithms [47], and maybe also several HNF-based encryption schemes since [40].

Decryption. We use the reduction algorithms we presented in this paper, which we proved to terminate. Note that in our proof-of-concept implementation, we used precomputations to avoid large integers and save time.

[4] Or use a permutation to attempt obtaining a perfect HNF as reported in [54].

4.4 Formal security

The scheme defined by the algorithms presented in Figure 1 is guaranteed to be correct but is not secure. Since it is deterministic, it is not even IND-CPA. In the following, we discuss the necessary milestones to reach in the path towards IND-CCA security. Note that, for example, the key encapsulation mechanism BAT [27] follows essentially the same steps.

One-wayness. The first level of security to achieve is one-wayness, i.e. that one cannot recover a message $\mathbf{m}$ given only the public key $\mathbf{P}_K$ and a random ciphertext $c = \texttt{DRE\_Encrypt}(\mathbf{P}_K, \mathbf{m})$. Obviously, an adversary is allowed to produce as many pairs of plaintext-ciphertext as he wants. Since this corresponds to solving a BDD on a diagonally dominant lattice (from one of its standard canonical forms), one-wayness is achieved under the conjecture that this problem is indeed hard to solve. As we will see later, recovering the secret key from its HNF can be reduced to several instances of BDD as well. Since nothing indicates that recovering $\mathbf{S}_K$ from $\mathbf{P}_K$ can be done in polynomial time, the one-wayness of DRE is a reasonable conjecture.

IND-CPA from one-wayness. Assume that the scheme achieve OW-CPA security. Then, following [25,27] one can be made IND-CPA security in the Random Oracle Model (ROM) with the following transformations of the encryption and decryption functions :

$$(\mathbf{m}, \mathbf{s}, \mathbf{P}_K, H) \mapsto [\mathbf{m} \oplus H(\mathbf{s}) \| \texttt{DRE-Encrypt}(\mathbf{P}_K, \mathbf{s})]$$

and

$$(\mathbf{c}_1, \mathbf{c}_2, \mathbf{S}_K) \mapsto H(\texttt{DRE-Decrypt}(\mathbf{S}_K, \mathbf{c}_2)) \oplus \mathbf{c}_1,$$

where $\mathbf{s}$ is a random vector and H a hash function modelized as a random oracle.

IND-CPA to IND-CCA Finally, famous transformations permit to reach IND-CCA security such as the Fujisaki-Okamoto (F.-O.) transform [28,17].

Remarks on one-wayness. In the end, we see that a last challenge would be to obtain a formally proven OW-CPA version of DRE. One option could be to adapt the proof from [27, Theorem 2] by considering a class of random co-cyclic lattices whose HNF are hard to distinguish from the ones of diagonally dominant matrices. We believe that this could be achieved through more extensive study of the determinant, which can be an easy sorting criterion.

We could also choose to hide the determinant, i.e. remove the last coefficient from the sent vector $\mathbf{h}$ before sharing it. In order to construct a setting where the public key is indistinguishable from random co-cyclic lattices, one could also randomize the public key by adding multiples of $\det(\mathbf{B})$ to its entries, so that it is close to uniform in a certain range $[\![2^{l-1}, 2^l]\!]$.

Note that it has been over 20 years that a similar structure, the GGH encryption of Micciancio [40] remains unbroken. We conjecture that the problem of distinguishing co-cyclic lattices with diagonally dominant bases (with similar parameters otherwise) from generic co-cyclic lattices is hard.

4.5 Concrete security

There are several security concerns that one needs to address if planning to build a cryptosystem. One of them is to ensure that deciphering **c** into **m** is not trivial without the secret key. Heuristically, if **c** is large enough, the problem of recovering **m** from **c** can be seen as a specific instance of the CVP, which is known to be hard. With that in mind, what is left is the security of the public key. Since [40], it makes sense to provide a basis of $\mathcal{L}(\mathbf{B})$ as a HNF for the public key, however other choices might be possible. It might not even be necessary to provide a basis of $\mathcal{L}(\mathbf{B})$ in the first place. Let us assume the public key is chosen as another basis of the same lattice: in the last decades, it seemed that pure key recovery attacks on diagonally dominant matrices [49,54] or close structures [30,43] are rather unsuccessful. The weaknesses were mostly on signature scheme instances [45,21,23] which do not concern this section. Note that [45] also consider that the *encryption* approach of [30] is still secure, and to the extent of our knowledge this claim has not been challenged yet.

Key recovery

Naive attack. The most naive attack is to reduce the public key in order to recover the secret key or a basis with an equivalent quality. As a matter of fact, we will consider only the complexity of computing *one* short vector. In the case of DRS, it amounts to solve the SVP_γ for a small constant approximation factor. Note also that diagonally dominant lattices have unusually short vectors. Indeed, the secret key $\mathbf{B}$ is composed of vectors such that $D \leqslant \|\mathbf{B}_i\|_2 \leqslant \sqrt{2}D$ which is smaller than what is predicted by the Gaussian heuristic by a factor in $O(\sqrt{n})$. Thus, the situation is similar to what happens for the Hawk cryptosystem based on the $\mathbb{Z}^n$-LIP. Following the analysis done in [24], the required blocksize to recover a secret vector should satisfy $\sqrt{\beta/n} \approx \delta_\beta^{2\beta-n-1}$ with $\delta_\beta \approx (\beta/(2\pi e))^{1/2(\beta-1)}$ which gives $\beta \in O(n/2) + o(n)$.

Attack by BDD-uSVP. Apart from reducing the public key, one can use the fact that $\mathbf{B}$ is diagonally dominant. Indeed, each vector of the secret basis is then of the form $D \cdot \mathbf{e}_i + \mathbf{n}_i$ with $\|\mathbf{n}_i\|_1 < D$. Then solving a BDD instance with respect to $\mathcal{L}(\mathbf{B})$ and the target vector $D \cdot \mathbf{e}_i$ would yield the secret vector $\mathbf{B}_i$. The cost of such an attack – without any additional knowledge – can be estimated following [4,2]. It is mentioned in [23] that recovering $\mathbf{B}_i$ can be done with BKZ-β when

$$\sqrt{\beta/(n+1)} \cdot \|\mathbf{B}_i - D \cdot \mathbf{e}_i\| \approx \delta_\beta^{2\beta-n-1} \cdot D^{n/(n+1)}. \tag{5}$$

Once fix the smallest block β that BKZ can use to be potentially successful, one can use the conservator cost estimation used in [27] for BKZ at

$$16(n+1)2^{0.292\beta}. \tag{6}$$

Attack on sparse keys. This previous attack can be eventually generalized if the vector $\mathbf{B}_i$ is reasonably sparse. As sparsity offers more compact keys, this is indeed a property that one can observe in the parameter we propose in Table 1. If $\mathbf{B}_i$ is sparse, for example with only l values different from 0, then one can guess k number of 0 and then perform the same attack as previously described. The block β_k of the successful BKZ will then respect

$$\sqrt{\beta_k/(n-k+1)} \cdot \|\mathbf{B}_i - D \cdot \mathbf{e}_i\| \approx \delta_{\beta_k}^{2\beta_k - n-k-1} \cdot D^{n/(n+1-k)}. \tag{7}$$

Obviously, the cost of running BKZ-β_k will have to be multiply by the number of guess required to succeed

$$\frac{C_n^l}{C_{n-k}^l} 16(n-k+1)2^{0.292\beta_k}. \tag{8}$$

Therefore, we will use the k such that the cost of Equation 8 is minimal to evaluate the security of the secret key.

Message recovery For message recovery, one needs to compute $\mathbf{m}$ from c, where c corresponds to a vector $\mathbf{c} \equiv \mathbf{m} \bmod \mathcal{L}(\mathbf{B})$. Thus $\mathbf{v} = \mathbf{c} - \mathbf{m}$ is a lattice vector such that $d(\mathbf{c}, \mathbf{v}) = \|\mathbf{m}\|$. In our system, $\|\mathbf{m}\|$ is clearly superior that $\mathbf{B}_i$ and therefore the message recovery will be always more costly than the key recovery. Consequently, it will not be useful to evaluate security.

4.6 Parameters and performance for DRE

We can obtain parameters for our DRE scheme together with the corresponding security levels. The latter are evaluated through the concrete cryptanalysis that we describe in Section 4.5, which provides us with optimal parameters as well. However, we choose to follow a linear dependency between them in order to simplify the presentation and obtain an additional margin of security. In the end, for a given security level λ, we have the following :

- the noise level $\nu(\mathbf{B})$ of the secret key is chosen to be $5\lambda/4$;
- the dimension n equals to $12 \cdot \delta(\mathbf{B})$;
- the diagonal coefficient D is then deduced from $\delta(\mathbf{B})$ following the conditions described in Section 4.2.

As a proof of concept, we also implemented our encryption scheme in order to verify that our theory is valid and hopefully that it could obtain reasonable performances without aggressive optimizations. Indeed, since it uses lattices with no polynomial structure, one could wonder to what extent how slow a basic implementation of DRE would be. We provide timings (in number of cpucycles) from our basic implementation, and the associated parameters in table 1

As comparison, these performance numbers are less than 10 times larger than the ones reported for the optimized implementation of FrodoKem[11], a third round NIST candidate. The code is available upon request, and will be made public upon publication.

Table 1: Parameters for DRE and performance (cpucycles)

Security level λ	32	64	128	192	256	512
Dimension n	480	960	1920	2880	3840	7680
Noise level	40	80	160	240	320	640
Diagonal D	123	243	483	723	963	1923
Encryption (avg cpucycles)	-	-	570,734	1,213,811	2,502,482	-
Decryption (avg cpucycles)	-	-	8,971,801	21,357,971	41,899,002	-

5 Heuristic patch of the DRS scheme

In this section we look into the impact of Ducas and Yu statistical attack [23] on the DRS signature scheme [49,55] when using parameters from our DRE scheme.

5.1 Quick recap of the DRS scheme and attacks

The signature scheme called DRS was a submission to the first round of the NIST standardization process for quantum resistant signature schemes [46], using diagonally dominant lattices. The main idea of DRS is to follow a framework close the one of GGH [31] but using the diagonal dominance property to sign within an hypercube independent of the secret key, hoping to prevent leaking the secret key as in [45] for example. This was first presented by Plantard et al. in [50]. However the original DRS scheme has been subject to a learning attack from Ducas and Yu [59], which was then extended to the second version of the scheme, the so-called DRSv2 [56]) [23].

The main idea behind this learning attack is that a signature $\mathbf{s}$ obtained from the signature algorithm is of the form $\mathbf{s} = \mathbf{s}' \pm \mathbf{B}_i$, where $\mathbf{B}$ is the secret diagonally dominant matrix and $\mathbf{s}'$ is the vector we have just before the algorithm stops. This relation introduces a correlation between the coefficients of the row $\mathbf{B}_i$ and the ones of $\mathbf{s}$. Then by collecting lots of signatures and using learning techniques, one can make an educated guess on a key. Typically, for the ith basis vector, one guess $\mathbf{B}'_i$ which is *close* to the secret $\mathbf{B}_i$.

In the following, we study a new version of the DRSv2 scheme where we use the same parameters as for our DRE scheme (see Section 4), except for D which is $\nu(\mathbf{B}) + 1$ as in the previous DRS schemes. In particular, we wish to evaluate the performance of the attack by Ducas and Yu for these new parameters.

5.2 Learning attack with new parameters

In order to evaluate the performance of their attack for a given dimension n and N samples, Ducas and Yu introduce the following factor :

$$r(n, N) = \frac{1}{n} \sum_{i=1}^{n} \frac{\|\mathbf{B}_i - \mathbf{B}'_i\|_2}{\|\mathbf{B}_i - D \cdot \mathbf{e}_i\|_2}.$$

This factor is deeply related to the complexity of the BDD-uSVP attack with target $\mathbf{B}'_i$ as it quantifies its distance to the secret vector $\mathbf{B}_i$. More precisely, if the learning attack gives us a given factor $r(n, N)$, we then know that we can safely replace $D \cdot \mathbf{e}_i$ by a target vector $\mathbf{t}_i$ such that $\|\mathbf{t}_i - \mathbf{B}_i\|_2 \sim r(n, N) \cdot \|D \cdot \mathbf{e}_i - \mathbf{B}_i\|_2$. Thus the smaller $r(n, N)$ is, the easier the subsequent BDD-uSVP procedure is.

In Figure 2 we plotted the factors $r(n, N)$ that we obtained using our new parameters for the DRS scheme.

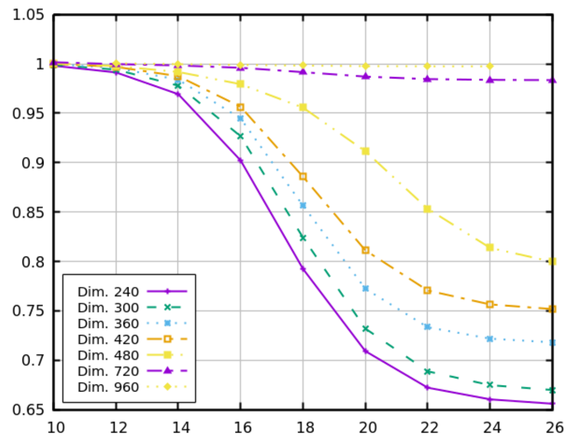

Fig. 2: New experimental measures of $r(n, N)$.

One can observe that with the chosen parameters and a fixed N, the factor $r(n, N)$ seems to be larger when n is increasing. This is different from the original DRSv2 scheme for which the statistical attack of Ducas and Yu becomes more and more efficient as n is increasing. Moreover, for all dimensions n the attacks seems to quickly stabilize with increasing sample size N. Again this is new when compared to the DRSv2 scheme. Finally, the attack produces almost no improvement for larger dimensions. All of these observations tend to show that the statistical attack *à la* Ducas and Yu [23] should not have an important impact on the asymptotical security of the DRS scheme with our new parameters.

5.3 Security analysis

As we consider a modified version of the DRSv2 scheme with the parameters as described above. Since this is the only major modification, we will not describe algorithms in detail. We focus on analysing its security. We refer to Section 4.5 for key recovery attacks from DRE which will work the same way here. Thus, we only consider more advanced techniques through statistical analysis.

Original attack. The first statistical attack from Nguyen and Regev [45] and its improvements [21,37] assume at some point that signatures are of the form

$\mathbf{s} = [s_1, \ldots, s_n] \cdot \mathbf{B}$ where the coordinates s_i are independent one to each other. There is no evidence that this condition is satisfied by DRS signatures. However, their distribution may be close to this ideal setting to the point where one can still apply the gradient descent with success. Moreover, remark that we know broad directions for the secret vectors $\mathbf{B}_i$. Thus, as mentioned by Nguyen and Regev for the GGH scheme in [45], one can start the descent with well-chosen initial vectors instead of drawing them uniformly on the unitary sphere. However, our experiments show that this strategy is asymptotically unsuccessful. Indeed, if $\mathbf{s}$ is a vector recovered by a descent, our experiments show that its distance to the secret key $\min_{i \in [\![1,n]\!]} \|\mathbf{s} - \mathbf{B}_i\|_2$ is typically around D. Thus, the best strategy remains the BDD-uSVP attack on $D \cdot \mathbf{e}_i$.

Learning attack from Ducas and Yu. [23] The data gathered by our experiments tend to show that the learning attack from Ducas and Yu is mitigated. But let us look more precisely into how the new factors $r(n, N)$ translate in terms of security. Recall that for a given $r(n, N)$, the key recovery can be done by replacing the target vector $D \cdot \mathbf{e}_i$ by a vector $\mathbf{t}_i$ such that $\|\mathbf{t}_i - \mathbf{B}_i\|_2 \geqslant r(n, N) \cdot \|D \cdot e_i - \mathbf{B}_i\|_2$. Thus the complexity is deeply connected to the distance $r(n, N) \cdot \|D \cdot e_i - \mathbf{B}_i\|_2$. In Table 2, for several security levels λ, we gather the theoretical distance $\|\mathbf{t}_i - \mathbf{B}_i\|_2$ under which the BDD-uSVP strategy starts to have a complexity lower than 2^λ versus the average distance that we obtained in our experiments.

Table 2: Targeted versus experimental distances for the learning attack.

Security level λ	32	48	64
Minimal distances for a successful attack	2.03	2.25	2.5
Experimental distances obtained	5.26	7.9	9.28

One can see that the distances that we obtained is way larger than the limit that we should not pass and this gap is increasing with the security level. Thus we deem that the learning attack should have minimal impact on the security of the scheme.

Extending the learning attack. However one can wonder whether signing with very close vector reveals other information, such as (potentially approximate) Voronoi cells. This lead us to consider the setting of the Closest Vector Problem with Preprocessing (CVPP).

Assume that a learning attack *à la* Nguyen and Regev [45,21,37] allows us to recover vectors from a hidden parallelotop close to the Voronoi cell. First one may wonder if this structure is complex enough to hide the secret basis. Indeed, diagonally dominant matrices have a strong structure allowing for an efficient CVP solver.

Thus we considered the possibility that the recovered vector could help in solving CVP_γ more efficiently, to the point where one could forge a signature in polynomial time. As mentioned earlier, this setting is close to the one of CVPP algorithms. We established in Appendix B that the average approximation factor reached by Algorithm 1, both for signed or negative noises, is a small constant. Following [20] the query phase for solving such an instance of the Approximate Closest Vector Problem with Preprocessing (CVPP_γ) is exponential for arbitrary lattices. Note that the size of the preprocessed list of lattice vectors should be (at least) subexponential as well and requires to compute the shortest vectors (up to some approximation factor) of the lattice, among which are the vectors of the secret basis. Thus, one would certainly recover the secret basis as a byproduct of the query phase. Thus, we deem that forging a signature using (approximate) Voronoi cells or classical algorithms solving the CVPP_γ [20] is as hard as recovering the secret key.

5.4 Performances and tested parameters

As in Section 4.6 for DRE, we implemented our version of the DRSv2 scheme using the same functions as in DRE to verify that its performance remain acceptable. Timings (in number of cpucycles) can be found in Table 3.

Table 3: Parameters for new DRS and performances (cpucycles)

Security level λ	128	256	512
Dimension n	1760	2640	3520
Noise level	160	240	320
Diagonal D	161	241	321
Signature (avg cpucycles)	8,369,109	17,848,387	31,778,217
Verification (avg cpucycles)	839,933	1,663,577	2,862,856

As a reference, this first proof-of-concept implementation of this new signature scheme is less than 3 times slower than the one submited as a candidate for the NIST PQC standardization by Squirrels [26], while our verification is less than 6 times slower. While the code is suboptimal and very basic, we believe it performs well enough to claim the approach is not to be discarded. The code is available upon request, and will be made public upon publication.

References

1. Ajtai, M.: The shortest vector problem in l_2 is NP-hard for randomized reductions. In: Proceedings of the thirtieth annual ACM symposium on Theory of computing. pp. 10–19. ACM (1998)
2. Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the Expected Cost of Solving uSVP and Applications to LWE. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 297–322. Springer International Publishing, Cham (2017)

3. Alekhnovich, M., Khot, S., Kindler, G., Vishnoi, N.: Hardness of approximating the closest vector problem with pre-processing. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05). pp. 216–225 (2005). https://doi.org/10.1109/SFCS.2005.40
4. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - a new hope. In: 25th {USENIX} Security Symposium ({USENIX} Security 16). pp. 327–343 (2016)
5. Babai, L.: On lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
6. Bajard, J.C., Imbert, L., Plantard, T.: Modular number systems: Beyond the mersenne family. In: International Workshop on Selected Areas in Cryptography. pp. 159–169. Springer (2004)
7. de Barros, C.F., Schechter, L.M.: GGH may not be dead after all. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics **3**(1) (2015)
8. Bauch, J., Bernstein, D.J., de Valence, H., Lange, T., Van Vredendaal, C.: Short generators without quantum computers: the case of multiquadratics. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 27–59. Springer (2017)
9. Bernard, O., Lesavourey, A., Nguyen, T.H., Roux-Langlois, A.: Log-S-unit Lattices Using Explicit Stickelberger Generators to Solve Approx Ideal-SVP. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology – ASIACRYPT 2022. pp. 677–708. Springer Nature Switzerland, Cham (2022)
10. Bernard, O., Roux-Langlois, A.: Twisted-phs: Using the product formula to solve approx-svp in ideal lattices. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 349–380. Springer International Publishing, Cham (2020)
11. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1006–1018. ACM (2016)
12. Brualdi, R.A.: Matrices eigenvalues, and directed graphs. Linear and Multilinear Algebra **11**(2), 143–165 (1982)
13. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: ASIACRYPT 2011. pp. 1–20. Springer (2011)
14. Conway, J., Parker, R., Sloane, N.: The covering radius of the leech lattice. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences pp. 261–290 (1982)
15. Cramer, R., Ducas, L., Peikert, C., Regev, O.: Recovering short generators of principal ideals in cyclotomic rings. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 559–585. Springer (2016)
16. Cramer, R., Ducas, L., Wesolowski, B.: Mildly short vectors in cyclotomic ideal lattices in quantum polynomial time. J. ACM **68**(2) (Jan 2021). https://doi.org/10.1145/3431725, https://doi.org/10.1145/3431725
17. Dent, A.W.: A designer's guide to kems. In: Paterson, K.G. (ed.) Cryptography and Coding. pp. 133–151. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
18. Dinur, I.: Approximating SVP_∞ to within almost-polynomial factors is NP-hard. In: Bongiovanni, G., Petreschi, R., Gambosi, G. (eds.) Algorithms and Complexity. pp. 263–276. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
19. Dopico, F.M.: Diagonally dominant matrices: Surprising recent results on a classical type of matrices (2014), https://gauss.uc3m.es/fdopico/talks/2014-manch-nasc.pdf

20. Doulgerakis, E., Laarhoven, T., de Weger, B.: Finding closest lattice vectors using approximate voronoi cells. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 3–22. Springer International Publishing, Cham (2019)
21. Ducas, L., Nguyen, P.Q.: Learning a zonotope and more: Cryptanalysis of NTRUsign countermeasures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 433–450. Springer (2012)
22. Ducas, L., van Woerden, W.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 643–673. Springer (2022)
23. Ducas, L., Yu, Y.: Learning strikes again: The case of the drs signature scheme. Journal of Cryptology **34**(1), 1–24 (2021)
24. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.: Hawk: Module lip makes lattice signatures fast, compact and simple. In: ASIACRYPT 2022. Springer-Verlag (2022)
25. Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A thorough treatment of highly-efficient ntru instantiations. In: Boldyreva, A., Kolesnikov, V. (eds.) Public-Key Cryptography – PKC 2023. pp. 65–94. Springer Nature Switzerland, Cham (2023)
26. Espitau, T., Niot, G., Sun, C., Tibouchi, M.: Squirrels: Square unstructured integer euclidean lattice signature (2023), https://squirrels-pqc.org
27. Fouque, P.A., Kirchner, P., Pornin, T., Yu, Y.: Bat: Small and fast kem over ntru lattices. IACR Transactions on Cryptographic Hardware and Embedded Systems **2022**(2), 240–265 (Feb 2022). https://doi.org/10.46586/tches.v2022.i2.240-265, https://tches.iacr.org/index.php/TCHES/article/view/9487
28. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology **26**(1), 80–101 (2013)
29. GERSGORIN, S.: Uber die abgrenzung der eigenwerte einer matrix. Bulletin de l'Academie des Sciences de l'URSS. Classe des Sciences Mathematiques et na **6**, 749–754 (1931)
30. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: CRYPTO'97, pp. 112–131. Springer (1997)
31. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski, B.S. (ed.) Advances in Cryptology — CRYPTO '97. pp. 112–131. Springer Berlin Heidelberg, Berlin, Heidelberg (1997)
32. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Algorithmic number theory, pp. 267–288. Springer (1998)
33. Klein, P.: Finding the closest lattice vector when it's unusually close. In: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms. pp. 937–941. Society for Industrial and Applied Mathematics (2000)
34. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (1982)
35. Lesavourey, A., Plantard, T., Susilo, W.: On ideal lattices in multicubic fields (2019), http://nutmic2019.imj-prg.fr/confpapers/MultiCubic.pdf
36. Limebeer, D.J.: The application of generalized diagonal dominance to linear system stability theory. International Journal of Control **36**(2), 185–212 (1982)
37. Lin, X., Suzuki, M., Zhang, S., Espitau, T., Yu, Y., Tibouchi, M., Abe, M.: Cryptanalysis of the Peregrine Lattice-Based Signature Scheme. Cryptology ePrint Archive, Paper 2023/1628 (2023), https://eprint.iacr.org/2023/1628, https://eprint.iacr.org/2023/1628

38. Merkle, R., Hellman, M.: Hiding information and signatures in trapdoor knapsacks. IEEE transactions on Information Theory **24**(5), 525–530 (1978)
39. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective. The Springer International Series in Engineering and Computer Science, Springer US (2012)
40. Micciancio, D.: Improving lattice based cryptosystems using the hermite normal form. In: International Cryptography and Lattices Conference. pp. 126–145. Springer (2001)
41. Micciancio, D., Goldwasser, S.: Complexity of lattice problems: a cryptographic perspective, vol. 671. Springer Science & Business Media (2012)
42. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Post-quantum cryptography, pp. 147–191. Springer (2009)
43. Micciancio, D., Warinschi, B.: A linear space algorithm for computing the Hermite normal form. In: Proceedings of the 2001 international symposium on Symbolic and algebraic computation. pp. 231–236. ACM (2001)
44. Minkowski, H.: Geometrie der Zahlen. B.G. Teubner, Leipzig (1896)
45. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. Journal of Cryptology **22**(2), 139–160 (2009)
46. NIST: Post-quantum cryptography standardization (2018), https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
47. NIST: Post-Quantum Cryptography: Digital Signature Schemes (2023), https://csrc.nist.gov/projects/pqc-dig-sig/standardization/call-for-proposals
48. Pellet-Mary, A., Hanrot, G., Stehlé, D.: Approx-SVP in ideal lattices with preprocessing. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 685–716. Springer (2019)
49. Plantard, T., Sipasseuth, A., Dumondelle, C., Susilo, W.: DRS : Diagonal dominant reduction for lattice-based signature. PQC Standardization Conference, Round 1 submissions (2018), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/DRS.zip
50. Plantard, T., Susilo, W., Win, K.T.: A digital signature scheme based on CVP max. In: PKC 2008. pp. 288–307. Springer (2008)
51. Regev, O.: New lattice-based cryptographic constructions. Journal of the ACM (JACM) **51**(6), 899–942 (2004)
52. Rump, S.M.: Estimates of the determinant of a perturbed identity matrix. Linear algebra and its applications **558**, 101–107 (2018)
53. Seo, E.Y., Kim, Y.S., Lee, J.W., Jong-Seon, N.: Peregrine: Submission to the korea post-quantum cryptography competition (2022), https://www.kpqc.or.kr/competition.html
54. Sipasseuth, A., Plantard, T., Susilo, W.: Enhancing Goldreich, Goldwasser and Halevi's scheme with intersecting lattices. Journal of Mathematical Cryptology **13**(3-4), 169–196 (2019)
55. Sipasseuth, A., Plantard, T., Susilo, W.: Improving the security of the DRS scheme with uniformly chosen random noise. In: Jang-Jaccard, J., Guo, F. (eds.) Information Security and Privacy. pp. 119–137. Springer International Publishing, Cham (2019)
56. Sipasseuth, A., Plantard, T., Susilo, W.: A noise study of the PSW signature family: Patching DRS with uniform distribution. Information **11**(3) (2020). https://doi.org/10.3390/info11030133, https://www.mdpi.com/2078-2489/11/3/133
57. Taussky, O.: A recurring theorem on determinants. The American Mathematical Monthly **56**(10P1), 672–676 (1949)

58. development team, T.F.: fpylll, a Python wrapper for the fplll lattice reduction library, Version: 0.6.0 (2023), https://github.com/fplll/fpylll, available at https://github.com/fplll/fpylll
59. Yu, Y., Ducas, L.: Learning strikes again: The case of the DRS signature scheme. In: ASIACRYPT 2018. pp. 525–543. Springer (2018)

A Diagonally Dominant with negative noise

One can obtain better results when considering more specific structures. In this section we consider diagonally dominant matrices $\mathbf{B} = \mathbf{D} + \mathbf{N}$ where the noise matrix $\mathbf{N}$ is such that $\forall (i,j) \in [\![1,n]\!], \mathbf{N}_{i,j} \leqslant 0$.

Lemma 3. *The bound on $\lambda_1^\infty(\mathcal{L})$ is tight, i.e. there is $\mathbf{B}$ such that $\lambda_1^\infty(\mathcal{L}(\mathbf{B})) = \Delta(\mathbf{B})$.*

Proof. Consider $\mathbf{B} = D \cdot \mathrm{Id}_n + \mathbf{N}$ such that $\mathbf{N}_{i,i+1} = 1 - D$ and $\mathbf{N}_{i,j} = 0$ whenever $j \neq i+1$. Then the vector $v \overset{\text{def}}{=} [1, \cdots, 1] \cdot \mathbf{B}$ satisfies the desired equality. □

Lemma 4. *Consider $\mathbf{B}$ a diagonally dominant matrix with negative noise. Then there is an algorithm – that we will denote by `neg-PSW` – that reduces any vector $\mathbf{v} \in \mathbb{R}_+^n$ to an equivalent vector $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(B)$ such that $\mathbf{w} \in [0, D[^n$.*

Proof. Let $\mathbf{v}$ be a vector and $\mathbf{w} \overset{\text{def}}{=} v - q \cdot \mathbf{B}_i$ for some $i \in [\![1,n]\!]$. Then remark that if $\mathbf{v}_i \geqslant qD$, we have $0 \leqslant \mathbf{w}_i < D$ and $\mathbf{w}_j \geqslant \mathbf{v}_j$ for all $j \neq i$. Moreover it is clear that $\|\mathbf{w}\|_1 = \|\mathbf{v}\|_1 - q\Delta(\mathbf{B})$. Thus it is clear that the algorithm will stop and that the outputted vector will lie in the claimed space. □

Remark 3. Note that one can easily shift the result to the centered hypercube $[\![-D/2, D/2[\![^n$ so that for any $\mathbf{v} \in \mathbb{N}^n$ there is $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(\mathbf{B})$ with $\mathbf{w} \in [\![-D/2, D/2[\![^n$.

One can note that the reduction radius is smaller (by a factor up to 2) that for generic diagonally dominant matrices. Moreover, the covering radius does no depends anymore of $\Delta(\mathbf{B})$. An advantage which can be used when diagonally dominant matrices are used for cryptography.

B Average quality of reduction

Average quality of CVP We evaluated experimentally the quality of the approximation factor obtained by Algorithm 1 as a CVP_γ solver for small dimensions. To this end we used the CVP solver from FPYLLL [58], called with the method `CVP.closest_vector(L, t)`, where `L` is the lattice and `t` is the target vector. From our computations, for a fixed dominance level $\Delta(\mathbf{B})$, the average approximation factor reached by Algorithm 1 is smaller than a constant, seemingly decreasing with respect to the dimension.

Note that since one is able to recover $\mathbf{B}$ from its HNF in exponential time, this indicates that approximating the CVPP within a small constant factor should be solvable in polynomial time for diagonally dominant matrices. This contrasts with the situation over general lattices [3].

	$\Delta(\mathbf{B})$ \ n	10	15	20	25	30	35	40	45	50	55	60
PSW	1	2.91	2.86	2.79	2.73	2.67	2.61	2.61	2.56	2.56	2.51	2.50
	$D/2$	1.55	1.50	1.50	1.37	1.38	1.41	1.43	1.36	1.38	1.36	1.38
neg-PSW	1	1.44	1.24	1.26	1.21	1.22	1.16	1.18	1.15	1.15	1.13	1.14
	$D/2$	1.066	1.022	1.028	1.015	1.021	1.012	1.018	1.010	1.011	1.010	1.009

Table 4: Average approximation factor reached by `PSW` and `neg-PSW` for small dimensions and $\Delta(\mathbf{B}) \in \{1, D/2\}$.

C Short vectors and reduction algorithms for Column Diagonally Dominant matrices

In this section we consider Column Diagonally Dominant matrices. A c.d.d matrices can be simply defined as the transpose matrix of a Diagonally Dominant matrix (Definition 9). Conceqeuntly, we will note $\Delta^T(\mathbf{B}) = \Delta(\mathbf{B}^T)$.

The overall methodology used in this subsection is very similar to the previous one. Again, the results proven in this subsection can be grouped in the following theorem.

Theorem 3. *Consider $\mathbf{B} \in \mathbb{Z}^n$ a c.d.d. matrix and $\mathcal{L} = \mathcal{L}(\mathbf{B})$. Then $\lambda_1(\mathcal{L}) \geqslant \Delta^T(\mathbf{B})$ and there is an algorithm, `RSR` (Alg. 3), running within a polynomial amount of arithmetic operations such that*

$$\forall \mathbf{v} \in \text{span}(\mathcal{L}), \texttt{RSR}(v) \equiv v \bmod \mathcal{L}, \|\texttt{RSR}(\mathbf{v})\|_\infty \leqslant D - \frac{\Delta^T(\mathbf{B})}{2}.$$

Consequently one has $\mu^{(\infty)}(\mathcal{L}) \leqslant D - \frac{\Delta^T(\mathbf{B})}{2}$.

As done previously, the proof of this theorem will be done in two steps: bounding the minimal size of the shortest vector, then bounding the maximal convergence radius of a reduction algorithm. Note that the acronym `RSR` stands for `RepeatedSingleReduce`.

C.1 Specific notations

We will use the following objects and notations.

- For $I \subset [\![1, n]\!]$, we denote by $\mathbf{B}_I \in M_{|I|,|I|}(\mathbb{Z})$ the submatrix of $\mathbf{B}$ composed of the rows and columns of indexes in I. Naturally, if $\mathbf{B}$ is a r.d.d/c.d.d matrix, so is $\mathbf{B}_I$.

– $S_\infty(l)$ is the set of positions i given $l \in \mathbb{Z}^n$ such that $|l_i| = \|l\|_\infty$

– $\mathcal{B}(I, \mathbf{B}) = \min \left\{ \max_{j \in I} \{ |(l \cdot \mathbf{B})_j| \mid \|l\|_\infty = 1, S_\infty(l) = I \} \right\}$ given any set of indexes I.

It is simply $\min\{\|l \cdot \mathbf{B}_I\|_\infty \mid l \in \{-1, 1\}^{|I|}\}$.

We denote $\mathcal{B}(I, \mathbf{B})$ by $\mathcal{B}_I$ when $\mathbf{B}$ is implied, and stress that $\mathcal{B}_I \neq \lambda_1(\mathbf{B})$.

C.2 Short vectors

First let us study the norm of a shortest vector.

Lemma 5 (Minimal largest value of non-zero combinations). *Consider $k \in \mathbb{Z}^n \setminus \{0\}$, $j \in [\![1, n]\!]$ such that $|k_j| = \|k\|_\infty$, $\mathbf{B}$ be a c.d.d matrix, and $\mathbf{v} = k \cdot \mathbf{B}$. Then one has $|\mathbf{v}_j| \geqslant \|k\|_\infty \cdot \delta_j(\mathbf{B}^T)$.*

Proof. Without any loss of generality we can assume $\mathbf{v}_i \geq 0$ and $k_j > 0$. Then

$$|\mathbf{v}_i| = \left| \sum_{i=1}^{n} k_i \mathbf{B}_{i,j} \right| \geqslant k_j D - \sum_{\substack{i=1 \\ i \neq j}}^{n} |k_i \mathbf{B}_{i,j}| \geqslant k_j (D - \sum_{\substack{i=1 \\ i \neq j}}^{n} |\mathbf{B}_{i,j}|) = k_j \delta_j(\mathbf{B}^T).$$

□

This directly implies that $\lambda_1^{(\infty)}(\mathcal{L}(\mathbf{B})) \geqslant \Delta^T(\mathbf{B})$. Let us show some additional results on c.d.d. matrices.

Lemma 6 (Submatrix bound on non-zero combinations). *Consider $\mathbf{B}$ a c.d.d. matrix, $k \in \mathbb{Z}^n$, $I = S_\infty(k)$ and $\mathbf{v} = k \cdot \mathbf{B}$. Then there is $j \in I$ such that $|\mathbf{v}_j| \geqslant \mathcal{B}(I, \mathbf{B})$.*

Proof. If $k \in \{-\|k\|_\infty, 0, \|k\|_\infty\}^n$, then there is $j \in S_\infty(k)$ such that $|\mathbf{v}_j| \geqslant \|k\|_\infty \times \mathcal{B}(S_\infty(k), \mathbf{B})$. If $\exists j_1, |k_{j_1}| \notin \{0, \|k\|_\infty\}$ with $k_{j_1} \neq 0$, one can pick j_1 such that $|k_{j_1}| \geqslant |k_j|$ for all $j \notin S_\infty(k)$. Consider the vectors k' and k'' such that $k = k' + k''$ and

$$k'_j = \begin{cases} \text{sign}(k_j) \cdot (|k|_\infty - |k_{j_1}|), & \text{if } j \in I \\ 0, & \text{otherwise.} \end{cases}$$

Therefore we also have

$$k''_j = \begin{cases} \text{sign}(k_j) \cdot |k_s|, & \text{if } j \in I \\ k_j, & \text{otherwise.} \end{cases}$$

Remark that for all $j \in S_\infty(k)$ we have $\text{sign}(k''_j) = \text{sign}(k'_j) = \text{sign}(k_i)$ and $|k''_j| = |k''|_\infty$. From what precedes we know that there is $j \in S_\infty(k)$ such that $|(k' \cdot \mathbf{B})_j| \geqslant \mathcal{B}(S_\infty(k), \mathbf{B})$. Moreover $S_\infty(k) \subset S_\infty(k'')$ and the signs are the same so $\text{sign}((k'' \cdot \mathbf{B})_j) = \text{sign}((k' \cdot \mathbf{B})_j)$. Thus we obtain $|(k \cdot \mathbf{B})_j| \geqslant \mathcal{B}(S_\infty(k), \mathbf{B})$. □

This gives us the following theorem.

Theorem 4 (Bound by the minimal submatrix). *Let $\mathbf{B}$ be a c.d.d. matrix. Then $\lambda_1^{(\infty)}(\mathcal{L}(\mathbf{B})) \geqslant \min_{I \subseteq [\![1, n]\!]} \mathcal{B}_I$.*

C.3 Reduction algorithms for c.d.d. matrices

The previous reduction algorithm only concerned r.d.d matrices and are not guaranteed to terminate on c.d.d matrices. We will propose here a different algorithm relying on the c.d.d structure. Before we present the full algorithm, we first introduce the core part that we denote by `SingleReduce`. It is described in Algorithm 2.

Algorithm 2 `SingleReduce`

Require: $\mathbf{v} \in \mathbb{Z}^n$, $\mathbf{B}$ a c.d.d matrix, $R_i \geqslant D - \frac{\delta_i(\mathbf{B}^T)}{2}$.
Ensure: $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(\mathbf{B})$ and $\|\mathbf{w}\|_\infty \leqslant \max(qR_i, \|\mathbf{v}\|_\infty - q\Delta^T(\mathbf{B}))$, where $q = \max\{t \in \mathbb{N}^* \mid \forall i \in [\![1, n]\!], \|\mathbf{v}\|_\infty - tR_i \geqslant t(\delta_i(\mathbf{B}^T))\}$

$w \leftarrow v$, $i \leftarrow 1$, $s \leftarrow [0, ..., 0] \in \{0,1\}^n$ {initialization vector, index, reduction status}
$q \leftarrow \max\{t \in \mathbb{N}^* \mid \forall i \in [\![1, i]\!], \|\mathbf{v}\|_\infty - tR_i \geqslant t(\delta_i(\mathbf{B}^T))\}$
while $\bigvee_{j=1}^n ((|\mathbf{w}_j| > qR_j) \wedge (s_j = 0))$ **do**
  if $|\mathbf{w}_i| > qR_i$ and $s_i = 0$ **then**
    $w \leftarrow w - q\frac{w_i}{|w_i|}\mathbf{B}_i$ {Reduce $|\mathbf{w}_i|$}
    $s_i \leftarrow 1$ {"Update" the reduction status of index i}
  end if
  $i \leftarrow (i \bmod n) + 1$ {Enforces i to be within $[1, n]$ and not $[0, n-1]$}
end while
return w

Lemma 7. *`SingleReduce` (Alg. 2) outputs $\mathbf{w} \in \mathbb{Z}^n$ verifying the following properties:*

1. $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(B)$.
2. $\forall i \in [\![1, n]\!], |\mathbf{v}_i| > qR_i \implies |\mathbf{w}_i| < |\mathbf{v}_i|$.
3. $\forall i \in [\![1, n]\!], |\mathbf{v}_i| \leqslant qR_i \implies |\mathbf{w}_i| \leqslant qR_i$.

Moreover the algorithm performs at most n additions on vectors.

Proof. First remark that we add or remove at most one time each row vector to the variable $\mathbf{w}$. This is ensured by the flag vector s. Therefore we add at most n vectors to $\mathbf{w}$. Write $\mathbf{v} = \mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(r)} = \mathbf{w}$ the two-by-two distinct values of the variable $\mathbf{w}$ with $r \leqslant n$. Similarly write $s^{(0)}, \ldots, s^{(r)}$ the different values taken by s. Fix some index $i \in [\![1, n]\!]$. First assume $s_i^{(r)} = 0$. Then we know that $|\mathbf{w}_i^{(r)}| \leqslant qR_i$ and w_i satisfies the claimed properties. Now assume $s_i^{(r)} = 1$. Let us denote by k_0 the integer such that $\mathbf{w}_i^{(k_0)} = \mathbf{w}_i^{(k_0-1)} \pm qD$. Without loss of generality we can assume $\mathbf{w}_i^{(0)} = \mathbf{v}_i \geqslant 0$. First we consider the case where $\mathbf{w}_i^{(0)} > qR_i$. Then for some $J \subset [\![1, n]\!] \setminus \{i\}$ we have

$$\mathbf{w}_i^{(k_0-1)} = \mathbf{w}_i^{(0)} + \sum_{j \in J} \pm q b_{j,i} \geqslant \mathbf{w}_i^{(0)} - q(D - \delta_i(\mathbf{B}^T)) > qR_i - q(D - \delta_i(\mathbf{B}^T)) \geqslant q\frac{\delta_i(\mathbf{B}^T)}{2} > 0$$

therefore $\mathbf{w}_i^{(k_0)} = \mathbf{w}_i^{(k_0-1)} - qD$. We can write

$$\mathbf{w}_i^{(n)} = \mathbf{w}_i^{(0)} - qD + \sum_{\substack{j \in [\![1,n]\!] \\ j \neq i}} \pm q b_{j,i} > qR_i - qD - q(D - \delta_i(\mathbf{B}^T) \geqslant -q(D - \frac{\delta_i(\mathbf{B}^T)}{2})$$

which ensures $|\mathbf{w}_i^{(n)}| < |\mathbf{w}_i^{(0)}|$. Now consider the case where $\mathbf{w}_i^{(0)} \leqslant qR_i$. From $D - \frac{\delta_i(B^T)}{2} > D - \delta_i(B^T)$ we deduce that $\mathbf{w}_i^{(k_0-1)} > 0$ and $\mathbf{w}_i^{(k_0)} = \mathbf{w}_i^{(k_0-1)} - qD$. With the same reasoning as before we can conclude $\mathbf{w}_i^{(n)} < \mathbf{w}_i^{(0)}$ and $\mathbf{w}_i^{(n)} > \mathbf{w}_i^{(k_0)} - qD - q(D - \delta_i(\mathbf{B})) > -q(D - \frac{\delta_i(\mathbf{B})}{2})$ which ensures $|w_i^{(n)}| \leqslant qR_i$. Finally we remark that the results obtained are independent of the choice of i. □

This building block naturally gives us the `RSR` reduction algorithm, which is guaranteed to finish given a c.d.d. lattice basis. Theoretically, there is no algorithm that can provide strictly better bounds on l_∞ for every single column diagonally dominant lattice: the covering radius cannot be lower than half the size of the shortest vector, and for $\Delta^T(\mathbf{B}) = D$ we do reach this extremity.

Algorithm 3 `RSR`

Require: $v \in \mathbb{Z}^n$, B a c.d.d matrix, $R_i \geqslant D - \frac{\delta_i(\mathbf{B})}{2}$.
Ensure: $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(\mathbf{B})$ and $|\mathbf{w}_i| \leq R_i$.
1: $\mathbf{w} \leftarrow \mathbf{v}$
2: **while** $\bigvee_{j=1}^{n}(|\mathbf{w}_j| > R_j)$ **do**
3: $\quad w \leftarrow$`SingleReduce`$(\mathbf{w},\mathbf{B},R)$.
4: **end while**
5: **return** w

Proposition 1. *Given a vector* $\mathbf{v} \in \mathbb{Z}^n$, $R \in \mathbb{Z}^n$ *such that* $R_i \geqslant D - \frac{\delta_i(\mathbf{B})}{2}$ *where* $D, \delta_i(\mathbf{B})$ *are associated to a c.d.d. matrix* $\mathbf{B}$*, RSR (Alg. 3) outputs* $\mathbf{w} \in \mathbb{Z}^n$ *verifying the following properties:*

1. $\mathbf{w} \equiv \mathbf{v} \bmod \mathcal{L}(\mathbf{B})$.
2. $\forall i \in [\![1,n]\!],\ |\mathbf{w}_i| \leqslant R_i$

Moreover the algorithm performs at most $n \left\lceil \log_b \frac{2\|\mathbf{v}\|_\infty}{2D+\Delta^T\mathbf{B}} \right\rceil + n$ *additions on vectors, where* $b = \frac{2D+\Delta^T\mathbf{B}}{2D-\Delta^T\mathbf{B}}$.

Proof. Consider $\|\mathbf{v}\|_\infty$ such that there is no integer $t > 0$ such that $\|\mathbf{v}\|_\infty - tR_i \geqslant t\delta_i(\mathbf{B})$, i.e. $\|\mathbf{v}\|_\infty - R_i < \delta_i(\mathbf{B})$. Then a call to `SingleReduce` with $q = 1$ outputs $\mathbf{w}$ such that $\|\mathbf{w}_i\| \leqslant R_i$. Now consider $\|\mathbf{v}\|_\infty$ sufficiently large so that q exists. One call to `SingleReduce` outputs $\mathbf{w}$ such that $\|\mathbf{w}\|_\infty \leqslant \max\{qR_i, \|\mathbf{v}\|_\infty - q\Delta^T(\mathbf{B}))\} \leqslant \|v\|_\infty - q\Delta^T(B))$ by definition of q. Thus we get $\|\mathbf{w}\|_\infty \leqslant \|\mathbf{v}\|_\infty \cdot$

$(1 - Q)$, where $Q = q\frac{\Delta^T(\mathbf{B})}{\|\mathbf{v}\|_\infty}$. Clearly $Q > 0$, and let us prove that $Q < 1$. By definition we have

$$\|\mathbf{v}\|_\infty - qR_i \geqslant q\delta_i(\mathbf{B}^T)) \implies \frac{q}{\|\mathbf{v}\|_\infty} \leqslant \frac{2D + \Delta^T(\mathbf{B})}{2}$$

which gives

$$Q \leqslant \frac{2\Delta^T(\mathbf{B})}{2D + \Delta^T(\mathbf{B})}.$$

Since $\Delta^T(\mathbf{B}) > 0$ one has $2D + \Delta^T(\mathbf{B}) > 2D$, which leads to $Q < 2D/2D = 1$.

Then, writing $a := 1 - \frac{2\Delta^T(\mathbf{B})}{2D+\Delta^T(\mathbf{B})} = \frac{2D-\Delta^T(\mathbf{B})}{2D+\Delta^T(\mathbf{B})}$ one has $0 < 1 - Q < a < 1$ and $\|\mathbf{w}\|_\infty \leqslant a \cdot \|\mathbf{v}\|_\infty$. Consequently, after i calls to `SingleReduce`, one has $\|\mathbf{w}\|_\infty \leqslant a^i \cdot \|\mathbf{v}\|_\infty$. Let us find i the number of calls to `SingleReduce` after which a single call to `SingleReduce` with $q = 1$ will output a well-reduced vector. This is ensured by

$$\begin{aligned}
\|\mathbf{w}\|_\infty \leqslant a^i \cdot \|\mathbf{v}\|_\infty < R + \Delta^T(\mathbf{B}) &\iff a^i \leqslant \frac{2D + \Delta^T(\mathbf{B})}{2\|\mathbf{v}\|_\infty} \\
&\iff i \geqslant \log_a \frac{2D + \Delta^T(\mathbf{B})}{2\|\mathbf{v}\|_\infty} \\
&\iff i = \left\lceil \log_a \frac{2D + \Delta^T(\mathbf{B})}{2\|\mathbf{v}\|_\infty} \right\rceil \\
&\iff i = \left\lceil \log_{1/a} \frac{2\|\mathbf{v}\|_\infty}{2D + \Delta^T(\mathbf{B})} \right\rceil.
\end{aligned}$$

Since each call to `SingleReduce` has at most n vector additions, we get the claimed worst-case cost. □

We want to stress this does not show the algorithm is practically efficient: `SingleReduce` might run a *quadratic* amount of absolute value comparisons on scalars in a single call. However, the reduction still runs a polynomial amount of vector operations in the dimension and in the entry size.

Comparison with Babai's Nearest Plane Unlike the r.d.d case, we do not have a measure of $\|b_i\|_1$. However, we estimate that it is possible in the case of c.d.d to have rows with very large noise, which might give $\|b_i\|_1 > 2D$ and thus a larger worst-case bound than a r.d.d for Babaï's nearest plane algorithm.