IACR Communications in Cryptology
ISSN 3006-5496, Vol. 2, No. 2, 35 pages.

https://doi.org/10.62056/ab0lmpgxq

Check for updates

Diagonally dominant matrices for cryptography

Andrea Lesavourey[1], Kazuhide Fukushima[2], Thomas Plantard[3] and Arnaud Sipasseuth[a,4]

[1] XLIM, Limoges, France
[2] KDDI Research, Fujimino, Japan
[3] Nokia Bell Labs, Murray Hill, NJ, USA
[4] University of Tokyo, Tokyo, Japan

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 DRE (Diagonal Reduction Encryption) as an application example: a decryption failure free encryption scheme using diagonally dominant matrices and provide an experimental implementation to prove its suitability as a research direction. The trapdoor neither uses floating point arithmetic nor polynomial rings, and yet is less than 10 times slower than other optimised unstructured lattice-based standardisation candidates. This work could apply to cryptosystems based on the Lattice Isomorphism Problem as well. As a bonus, we also 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 [Ger31, Bru82, Rum18]. Numerous applications of diagonal dominance can be found in various fields such as numerical linear algebra [Lim82], 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 applications in cryptography on few specific instances [PSDS18, SPS19b] where in both papers the focus was more 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 [GGH97, Mic01, PSW08] as well as in modular arithmetic [BIP04].

E-mail: andrea.lesavourey@unilim.fr (Andrea Lesavourey), ka-fukushima@kddi.com (Kazuhide Fukushima), thomas.plantard@nokia-bell-labs.com (Thomas Plantard), sipasseuth@g.ecc.u-tokyo.ac.jp (Arnaud Sipasseuth)

[a]This work was supported by JST CREST Grant Number JPMJCR2113, Japan.

[1]A history of this theorem through the ages can be seen in [Tau49]

[2][Dop14] lists some applications.

Received: 2025-04-08 Accepted: 2025-06-02

Euclidean lattices. The study of computational problems on lattices in general is also an old and very studied topic [Min96, CPS82, Bab86]. 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 (CVP) – can be proven to be NP-hard in the general case [Ajt98, MG12]. 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 [AKKV05]. The hardness of these problems over Euclidean lattices motivated cryptographers to consider them as building blocks for cryptographic schemes [GGH97, Reg03], 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 [GGH97] or constructions using the plain Learning With Errors (LWE) problem such as Frodo [BCD+16]. 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 noticeable constructions include NTRU [HPS98] and 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 [CDPR16] and even classical polynomial time over multiquadratic [BBdV+17] or multicubic fields [LPS20]. Then Cramer, Ducas and Wesolowski extended the analysis of [CDPR16] to the Ideal-SVP and showed that one could obtain a subexponential approximation factor in quantum polynomial time [CDW21]. With a slightly different approach, this result can be generalised to all number fields provided an exponential preprocessing phase [PHS19], which might be an artefact of the proof if we refer to experimental results obtained in [BLNR22, BR20]. 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. If an algebraic attack is discovered and the current schemes collapse, it is important that research have started exploring alternatives that would resist those attacks, *even if they end up being less efficient than the current schemes*. Beyond the pursuit of science and academic curiosity, starting research on potential backups is a reasonable safety measure. The NIST seems to have understood those last points very well: hence, they announced in their standardisation efforts to be interested in schemes *not based* on structured lattices [NIS23b]. Thus, studying other types of trapdoors or constructions is still an interesting and important research direction, recently explored in at least two NIST submissions: Squirrels [ENST23] or Hawk [DvW22, DPPvW22] for example. Note, that both of those are signature schemes.

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 [NR06]. 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 [DN12]. The same kind of attack [LSZ+24] has recently been applied to break the PEREGRINE signature scheme [SKLJS22]. In order to prevent [NR06]'s attack, Gentry, Peikert and Vaikuntanathan [GPV08] proposed a different signature procedure based on a variant of the Babai round-off algorithm [Kle00]. This so-called "GPV framework" is one of the most popular method to produce secure signatures and is used in the recently standardised schemes and most of the lattice-based candidates for standardisation.

The DRS signature scheme and diagonally dominant matrices A less popular method was also proposed by Plantard, Win and Susilo [PSW08].

They proposed a hash-then-sign procedure based on the infinity norm such that the signature space lie in a parallelepiped independent of the secret key. This is achieved using specific 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 standardisation [PSDS18], 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 [PSW08]. This scheme has known a learning attack by Ducas and Yu [DY21] showing that while the signature space seems to be indeed independent of the secret basis, a correlation still exists between the latter and the signature distribution. 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 [SPS20]. However, it remains a serious attack with around 30 bits of security loss for the first set of parameters, using 2^{30} signatures only.

Digital encryptions with lattices. On the other hand, statistical leaks tend to be a non-issue for encryption schemes. Indeed, the owner of the secret key do not need to send any public data apart from the public key, thus learning attacks like [NR06] do not apply in this case: [NR06] even states on the first few pages than the encryption version of GGH is probably still secure. More recent papers makes the same observation. [FKPY22] reminds for example that while the encryption of GGH is not broken, its security is not clearly understood. Knapsack problems, which can be considered specialised variants of lattice problems, also suffer from a bad reputation but recent papers such as [DvW22] do remind the academic community that not all knapsacks are completely broken. As far as we know, the encryption concept of GGH is still unbroken as of today. The current new standard for lattice-based encryption is KYBER [BDK+18], and while the NIST did not call for additional encryptions, the fact remains that KYBER share a similar structure to DILITHIUM [DKL+21], thus a new algebraic attack could affect them both. Novel encryption schemes, based on non-structured lattices, are thus also interesting.

1.2 Our Contributions

This work is composed of two 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 [Din00], thus we believe providing a tighter upper bound for

any specific family of lattices is an interesting result in itself if we ever wish to use this subfamily as a trapdoor for constructing cryptographic schemes, to at least measure the difference with other (subfamilies of) lattices. Then we give an improved study of the reduction algorithm of [PSW08] for diagonally dominant matrices and prove a stronger reduction capability than previously proven for such lattices [SPS19b]. 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 [GGH97, dBS15]. We discuss formal security and the steps to take towards Indistinguishable under Chosen Ciphertext Attack (IND-CCA) security, using standard techniques or transformations [FO99, Den03]. We also evaluate the practical security of the scheme using common cryptanalytic techniques to assess lattice-based constructions, and show that our construction is asymptotically secure.

We also provide in an appendix a reparametrisation of the Diagonal Reduction Signature (DRS) signature scheme using the work done on this paper, and further properties and the behaviour of the reduction algorithms over both generic and particular forms of diagonally dominant basis.

Conclusion and future works We show that an encryption version of DRS, that we named DRE, is possible and prove it through theoretical work and implementation. The experimental implementation shows that while the performance may not match the current alternative unstructured standard FrodoKem [BCD+16], the order of magnitude are still relatively close to each other, showing it is possible to have an encryption scheme based on lattices outside q-ary that do not rely on polynomial rings or floating point arithmetic. Furthermore, note that the main drawback of diagonally dominant matrices is that their structure allow for easier attacks on the secret key. This leads to increased parameter sets, e.g. the dimension, to reach similar security levels. A workaround could be to apply such matrices to the Lattice Isomorphism Problem (LIP) [DvW22, BGPS23] framework, where the "good basis" is public. Since the attack vectors on such constructions are different, see [DPPvW22], considerations such as hiding the underlying lattice are discarded, which could result in drastically increased efficiency compared to the encryption scheme we introduced in this paper. Finally using diagonally dominant matrices could be more conservative than $\mathbb{Z}^n$. Indeed, note that the identity matrix is diagonally dominant, but degenerated since it has no noise.

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 will also use the notation $\mathbf{1}_n$ for the all-ones vector $[1, \ldots, 1]$ in dimension n.

2.1 Cryptography and security notions

We recall standard security notions related to encryption schemes in asymmetric cryptography. In the following, we will write $\mathcal{M}$ and $\mathcal{C}$ for the sets of plaintexts and ciphertexts, respectively.

Definition 1. An *encryption scheme* is defined by a triplet $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ such that :

- KeyGen takes a security level λ as input, outputs a public/secret key pair $(\mathbf{P}_K, \mathbf{S}_K)$.
- Enc takes $\mathbf{P}_K$ and a plaintext $m \in \mathcal{M}$ as input, outputs a ciphertext $c \in \mathcal{C}$.
- Dec takes $\mathbf{S}_K$ and a ciphertext $c \in \mathcal{C}$ as input, outputs a plaintext $m \in \mathcal{M}$.

The encryption scheme is said to be *correct* if :

$$\forall m \in \mathcal{M}, \forall (\mathbf{P}_K, \mathbf{S}_K) \leftarrow \mathsf{KeyGen}(\lambda), \mathsf{Dec}(\mathbf{S}_K, \mathsf{Enc}(\mathbf{P}_K, m)) = m.$$

Apart from correctness, an encryption scheme can achieve different security guarantees. Those are determined by attack simulations, commonly seen as "games" where the adversary, without knowledge of $\mathbf{S}_K$, tries to win against the challenger, who owns $\mathbf{S}_K$. In all those games, it is assumed that $\mathbf{P}_K, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$ are available to all parties.

The first security guarantee is one-wayness, which is really the minimum required degree of security.

Definition 2 (The One-Way under Chosen Plaintext Attack (OW-CPA) game)**.**

1. Challenger gives $c \leftarrow \mathsf{Enc}(\mathbf{P}_K, m)$ for $m \leftarrow \mathcal{U}(\mathcal{M})$ to adversary.
2. Adversary wins the game by correctly guessing m.

The scheme is OW-CPA if no Probabilistic Polynomial Time Adversary (PPTA) can win the game with non-negligible probability.

In order to account for more complex attack scenarii, one needs to introduce more advanced security notions. The first security upgrade is to let the adversary choose between two choices.

Definition 3 (The Indistinguishable under Chosen Plaintext Attack (IND-CPA) game)**.**

1. Adversary generates two distinct plaintexts m_0, m_1 and send them to challenger.
2. Challenger computes $c = E(\mathbf{S}_K, m_b)$ for $b \in \{0, 1\}$
3. Adversary wins the game by correctly guessing b.

The scheme is IND-CPA if no PPTA can win the game with non-negligible probability.

Finally, one can give more power to the attacker, for example having access to a decryption oracle *before playing the IND-CPA game*. This leads to the notion of Indistinguishable under Chosen Ciphertext Attack (IND-CCA).

Definition 4 (The IND-CCA game)**.**

1. Adversary requests $m_i = \mathsf{Dec}(\mathbf{S}_K, c_i)$ for any number of chosen c_i, to which the challenger complies. Adversary stores the knowledge.
2. Adversary wins by winning a new IND-CPA game with the knowledge above.

A scheme is IND-CCA if no PPTA can win the game with non negligible probability.

It is possible to prove that a cryptosystem reach any of these three levels directly. However, it is also frequent to use classical transformations to transform an encryption scheme into a new one reaching a stronger security level. An important example is the Fujisaki-Okamoto (F.-O.) [FO99, Den03] transform allowing us to transform a IND-CPA scheme into a IND-CCA one.

2.2 Euclidean lattices

We refer readers to [MG12, MR09] for a more complete background of lattice theory.

Definition 5 (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 we denote by $\mathcal{L}(\mathbf{B})$ the lattice generated by the rows of a basis matrix $\mathbf{B}$. We write the *determinant* (or *volume*) 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, there is a standard shape, unique, computable in polynomial time from any basis and used in almost every modern lattice-based cryptosystem, which we define below:

Definition 6 (Hermite Normal Form (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. In particular, we pay attention to lattices which HNF have a special form:

Definition 7 (Perfect HNF)**.**
Let $\mathbf{H} \in M_{n,n}(\mathbb{Z})$ a matrix in HNF. We say $\mathbf{H}$ is a *perfect* HNF if, and only if, it admits Id_{n-1} as a submatrix: in other words, there is at most one non-trivial column vector.

While HNF have been considered in cryptography since [Mic01], *perfect* HNF have not always been considered since specific popular lattices arising from Learning With Errors (LWE) [Reg03] or "Number Theory is Really Useful?" (NTRU) [HPS98] *cannot* have such shapes (in practical instantiations). Interestingly, if a lattice admits a perfect HNF as a basis, then it is co-cyclic.

Definition 8 (Co-cyclic lattice)**.**
Let $\mathcal{L} \subset \mathbb{Z}^n$ be a full-rank lattice. Then $\mathcal{L}$ is *co-cyclic* if, and only if, $\mathbb{Z}^n/\mathcal{L}$ is cyclic.

For any sufficiently large dimension on any fixed maximal volume, co-cyclic lattices form a large class of *all* lattices ($\approx 85\%$ [NS16]). Not only those are more numerous, and likely more directly linked to generic problems beyond cryptography compared to q-ary lattices, they *also* benefit from an average-to-worst-case reduction [GINX16].

To each Euclidean lattice are attached important geometrical invariants, i.e. not linked to a specific representation, which are linked to fundamental problems on lattices. We consider two of them here, namely the *first minima* and the *covering radius*.

Definition 9 (Minima of a lattice)**.** We denote by $\lambda_k^{(l)}(\mathcal{L})$ the smallest value r such that a ball centred in zero and of radius r in norm l contains k linearly independent vectors of $\mathcal{L}$.

Definition 10 (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 $\|\mathbf{x}-\mathbf{v}\|_l < \mu^{(l)}(\mathcal{L})$.

Note that the previous definitions are dependent on the norm chosen to measure the ambiant space. Moreover, some relations exist between those quantities. For example, for any lattice we have $\frac{1}{2}\lambda_1^{(2)}(\mathcal{L}) \leq \mu^{(2)}(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n^{(2)}(\mathcal{L})$ (See [MG12]).

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

Definition 11 (Approximate Shortest Vector Problem (SVP_γ))**.** 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 $\|v\| \leqslant \gamma \cdot \lambda_1(\mathcal{L})$.

Definition 12 (Approximate Closest Vector Problem (CVP_γ))**.** 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 $\mu^{(l)}(\mathcal{L})$ offer some natural bounds to transform the problem Closest Vector Problem (CVP) in some useful variants, especially for cryptographic applications.

Definition 13 (γ-Guaranteed Distance Decoding (GDD_γ))**.** 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_γ, is key for lattice-based signature schemes, and the second variant below is key for lattice-based encryption schemes.

Definition 14 (Bounded Distance Decoding (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. A "good" basis is typically one composed of vectors that are as short and as orthogonal as possible: the better the basis is, the heuristically "easier" is it to solve the aforementioned lattice problems in general. On the other hand, a "bad" basis is one where the aforementioned problems are "harder" to solve with. Typically a LLL-reduced [LLL82] or BKZ-reduced [CN11], together with an appropriate algorithm such as Babai's round-off or nearest plane algorithms [Bab86]. For example, the approach proposed by Klein [Kle00] for solving BDD for some α. In general, those "good" basis are obtained with the algorithms of the same name, that takes any basis as an input and attempt to compute a "reduced" basis (i.e "good" enough). BKZ-β, where β is a block-size, is the usual reduction algorithm, where (grossly speaking) $\beta = 2$ is LLL, up to n for a HKZ-reduced lattice base. The higher the β, the "better" is the output basis, but the slower is the algorithm. The heuristic security of lattice-based schemes, whether based on GDD_γ or BDD, is "in general" evaluated on finding an appropriate β for which we can "break" the lattice-based cryptosystem, and extrapolating the minimum attack cost as the cost of running BKZ-β.

Remark 1. Note that a CVP_γ algorithm can be used as a GDD_γ solver as long as the approximation factor γ ensures that any target has a solution. Remark also that solving the GDD_γ is equivalent to computing a *short* coset representative of $\mathbf{t}$ mod $\mathcal{L}$. 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 13 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, leaving the HNF of the lattice as a "bad" basis to be the public key: the HNF as a public key was first proposed by Micciancio [Mic01], where the "good/bad" basis approach was proposed by [GGH97] which itself is a lattice derivative from the code-based McEliece [McE78].

Definition 15 (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 analyse 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 Section B 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.

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

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 [PSW08] and is a known approximation of Babai's Round-off algorithm [Bab86] 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 [PSDS18] within the DRS scheme. The algorithm was proven to terminate with $\|\mathbf{w}\|_\infty < D$ in [PSDS18], but did not take into account the leeway $\Delta(\mathbf{B})$. A slight modification of the reduction proof given in [SPS19b] gives us a tighter bound by changing the loop condition in line 2 of the algorithm to a comparison with a value $R_i \geqslant D - \delta_i(\mathbf{B})/2$ for every index i. This gives us the modified version, described in Algorithm 1.

[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.

Algorithm 1 *Modified* 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: w ← v
2: while ⋁_{j=1}^{n} (|w_j| > R_j) do
3:    i ← any index such that |w_i| > R_i
4:    if |w_i| ⩾ D then
5:       q ← sign(w_i) · ⌊|w_i| / D⌋
6:    else
7:       q ← sign(w_i)                    {Extra reduction not in original PSW}
8:    end if
9:    w ← w − q · B_i                     {Reduce |w_i|}
10: end while
11: return w
```

Correctness. The following lemma states that for a given R, the algorithm terminates given that the values of R_i are above a certain bound which varies for each index. Note that one could remove the reference to the quantities R_i and stick with the smaller $D - \delta_i(\mathbf{B})/2$. However we wants to stress that a user can fix at which level Algorithm 1 stops.

Lemma 1 (Tighther bound in PSW-reduction algorithm)**.** *For input $\mathbf{v} \in \mathbb{R}^n$, 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 $\mathbb{R}^n \times [\![1, n]\!]$ which acts on $\mathbf{w}$ as steps 4-9 of Algorithm 1. Thus, it can be described by $f : (\mathbf{w}, i) \mapsto \mathbf{w} - \text{sign}(\mathbf{w}_i) \cdot \left(\left\lfloor \frac{\mathbf{w}_i}{D} \right\rfloor + \chi_{]R_i, D[}(\mathbf{w}_i) \right) \cdot \mathbf{B}_i$, where χ_A designates the indicator function of a given set A. In order to show that Algorithm 1 ends and outputs a correct vector, we will prove the following:

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

for some fixed value $\varepsilon > 0$. 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 as $\mathbf{w}_i$, 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

$$\|f(\mathbf{w}, i)\|_1 \leqslant \|\mathbf{w}\|_1 + \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot \delta_i(\mathbf{B}) \leqslant \|\mathbf{w}\|_1 - \left\lfloor \frac{|\mathbf{w}_i|}{D} \right\rfloor \cdot \delta_i(\mathbf{B}) < \|\mathbf{w}\|_1 - 1.$$

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 $t \in]\!]0, D - R_i[\![$, 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

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

Since the value $r_i \in [0, 1[$ such that $r_i \equiv \mathbf{w}_i \bmod 1$ does not change throughout the algorithm, t can take a finite number of values (at most $2n$). Denoting by ϵ the minimal of those values and taking ε as the minimum between 1 and ϵ, we obtain that (1) is indeed true. □

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 Babai's nearest plane algorithm which needs the GSO or Babai'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 time complexity of Algorithm 1 was briefly experimented in [PSW08], however a proper worst-case analysis was not provided and does not seem to have been done in the literature. We give a tighter analysis when the decoding radius R_i is chosen to be the smallest possible and the input vector $\mathbf{v}$ is in $\mathbb{Z}^n$.

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 $\frac{2nD}{2nD - \Delta(\mathbf{B}))}$. Consider also that for each $1 \leqslant i \leqslant n$, R_i is equal to $D - \delta_i(\mathbf{B})/2$. An upper bound on the complexity of vector operations done by Algorithm 1 is in*

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

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 $O\left(\log_b\left(\frac{\|\mathbf{v}\|_1}{2nD}\right)\right)$ vector 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 impactful possible reduction of $\|\mathbf{w}\|_1 \leqslant nD$ per step until $\|\mathbf{w}\|_1 = 0$: each step reduces $\|\mathbf{w}\|_1$ of at least $2t$, where $t = \mathbf{w}_i - R_i$. Since $\mathbf{w}_i \in \mathbb{Z}^n, \mathbf{B} \in M_n(\mathbb{Z})$ and $R_i = D - \delta_i(\mathbf{B})/2$, we know that t is at least $1/2$ so $2t$ is greater than 1. Therefore, we upper-bound the amount of loop iterations left by $\|\mathbf{w}\|_1 \leqslant nD$.

□

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 approximation 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 [PSDS18, SPS19b, SPS20], 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 [PSW08, SPS19b, PSDS18], 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 strong link between $\Delta(\mathbf{B})$ and the quality of reduction modulo $\mathcal{L}(\mathbf{B})$ has been described in the previous section. In this section, we present a second result linking once again $\Delta(\mathbf{B})$ with the first minimum of the underlying lattice. To our knowledge, this is the first result in this direction. Moreover, as we will see in Section 4, this permits the use of diagonally dominant matrix for encryption, even avoiding any 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 $\|\mathbf{v}\|_\infty \geqslant \Delta(\mathbf{B})$. First, we bound the taxicab norm using a classic norm inequality

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

Then, 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, that we call DRE as a callback to DRS [PSDS18], a signature scheme we mentioned in the introduction. The following encryption scheme will follow both similar lattice structures (diagonally dominant matrices) and associated algorithms (variants of [PSW08]). 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 $(D, n, M) \in \mathbb{N}^3$. Let us denote $\mathcal{L}$ the lattice generated by a diagonally dominant matrix $\mathbf{B} = D \cdot \mathrm{Id}_n + \mathbf{N}$. Let $R \in \mathbb{R}$ be a radius for which for any $\mathbf{c} \in \mathbb{Z}^n$ we can compute a vector $\mathbf{m} \equiv \mathbf{c} \in \mathcal{L}$ s.t. $\|\mathbf{m}\|_\infty < R$. Algorithms 1 and 3 offer 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 $\texttt{Encrypt}(\mathbf{P}_K, \mathbf{m}) = s \cdot \mathbf{H} + \mathbf{m}$, with $s \in \mathbb{Z}^n$.
- The decryption function will be $\texttt{Decrypt}(\mathbf{S}_K, \mathbf{c}) = \texttt{Reduce}(\mathbf{B}, \mathbf{c})$, 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 $\texttt{Reduce}(\mathbf{B}, \mathbf{c})$ is indeed a valid message. Then one needs to ensure unicity, meaning $\texttt{Reduce}(\mathbf{S}_K, \texttt{Encrypt}(\mathbf{P}_K, \mathbf{m})) = \mathbf{m}$. 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) is satisfied for matrices $\mathbf{B}$ such that

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

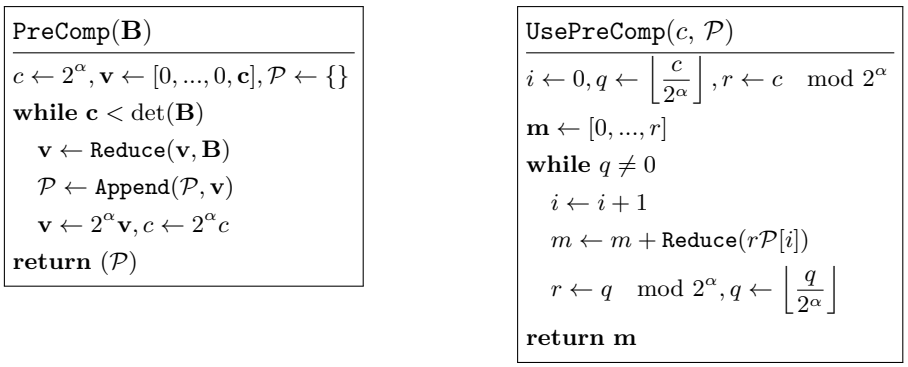

Figure 1: Generic Large Reduction Preprocessing, with arbitrary binary power base α

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 $\Delta(\mathbf{B})$. A fully mono-signed noise is explored in appendix.

4.3 Instantiation of the encryption scheme

To instantiate our encryption scheme, we first need to fix some public parameters such that the diagonal coefficient D and the dimension n. We assume the message space is composed of vectors in $\mathcal{M} = \{-1, 0, 1\}^n$, but we showed earlier that could also be subject to change. We fix some notations to ease the understanding of the scheme:

- $(D, n, c) \in \mathbb{N}^3$ are respectively the diagonal value, dimension and ciphertext.
- $\mathbf{H}, \mathbf{B} \in M_n(\mathbb{Z})$, and $\mathcal{P}$ is rectangular but has n columns.
- `IsNotPerfect` outputs `TRUE` anytime the input is not a perfect HNF.
- `RDDgen`$(D, n, \Delta(\mathbf{B}))$ is the random generation of a diagonally dominant matrix of size $n \times n$ which respects the diagonal D and the noise gap $\Delta(\mathbf{B})$.
- `PreComp` takes a diagonally dominant matrix and precompute a set of short coset representative modulo $\mathcal{L}(\mathbf{B})$, produced by `PSW2`. Those cosets are the ones of the form $[0, \ldots, 0, a]$, with a being large integers.
- `UsePreComp` uses the above to output a prereduction of the vector $[0, \ldots, 0, c]$ modulo $\mathcal{L}(\mathbf{B})$.

Note that `PreComp` and `UsePreComp`, that we present briefly in Figure 1, 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.

Using those notations, we present a high-level description of the instantiated encryption scheme in Figure 2.

From an external point of view, our scheme is based on a knapsack problem, such as the first proposition of Merkle-Hellman [MH78]. The major differences with Merkle-Hellman lie within the setup, the decryption and the trapdoors. Indeed, both public keys are a single series of integers but our construction is based on Euclidean lattices which were

DRE-KeyGen$(D,\ n)$

$\Delta(\mathbf{B}) \leftarrow 2(D+1)/3$
$\mathbf{H} \leftarrow \mathbf{0}, \mathbf{B} \leftarrow \mathbf{0}$
while IsNotPerfect$(\mathbf{H})$
  $\mathbf{B} \leftarrow$ RDDgen$(D, n, \Delta(\mathbf{B}))$
  $\mathbf{H} \leftarrow \mathrm{HNF}(\mathbf{B})$
$\mathcal{P} \leftarrow$ PreComp$(\mathbf{B})$
$\mathbf{h} \leftarrow \mathbf{H}[1..n, n]$
return $(\mathbf{P}_K = \mathbf{h}, \mathbf{S}_K = \mathbf{B}, \mathcal{P})$

DRE-Encrypt$(\mathbf{P}_K,\ \mathbf{m})$

$c \leftarrow 0$
for $i \in [1, n-1]$
  $c \leftarrow c - \mathbf{m}_i \cdot \mathbf{h}_i$
$c \leftarrow (c + \mathbf{m}_n) \bmod \mathbf{h}_n$
return c

DRE-Decrypt$(\mathbf{S}_K,\ \mathcal{P},\ c)$

$\Delta(\mathbf{B}) \leftarrow 2(D+1)/3$
$R \leftarrow (D - \Delta(\mathbf{B})/2) \cdot \mathbf{1}_n$
$\mathbf{m} \leftarrow$ UsePreComp$(c, \mathcal{P})$
$\mathbf{m} \leftarrow$ Reduce$(m, \mathbf{B})$
return m

Figure 2: DRE Algorithms

not in the mind of Merkle and Hellman when they proposed their scheme. Note that the Squirrels [ENST23] submission is also similar, although it is a signature scheme: a single series of large integers as a public key, although those are decomposed into small integers through the Chinese Remainder Theorem using a friable determinant. The concrete security, as we explore in Section 4.5, is measured by the currently best attacks on knapsacks: lattice-based reduction.

4.3.1 Key generation.

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 $\Delta(\mathbf{B}) = 2(D+1)/3$. 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 $\mathbf{H} = \mathtt{HNF}(\mathbf{B})$ but since it holds a perfect form, only its unique dense column vector $\mathbf{h}$ needs to be sent. We also perform precomputations to ease future decryptions, but in theory this is not needed.

4.3.2 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 as shown in Figure 2 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

$$\left[\begin{array}{ccccc} m_1 & \ldots & \ldots & m_{n-1} & m_n \\ \hline 1 & 0 & \ldots & 0 & h_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \ldots & 0 & 1 & h_{n-1} \\ 0 & \ldots & \ldots & 0 & \det(\mathbf{B}) \end{array}\right],$$

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

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

[4]Or use a permutation to attempt obtaining a perfect HNF as reported in [SPS19a].

Note that this approach is very similar to the one chose in the SQUIRRELS scheme [ENST23] recently submitted to the NIST call for proposals for quantum-resistant digital signature algorithms [NIS23a], and maybe also several HNF-based encryption schemes since [Mic01].

4.3.3 Decryption.

We use Algorithm 1, which we proved to terminate. Note that in our proof-of-concept implementation, we used precomputations to avoid large integers and save time.

4.4 Formal security

The scheme defined by the algorithms presented in Figure 2 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 path towards IND-CCA security. We first prove that our scheme is OW-CPA relying on the conjectured hardness of a distinguishing problem, then use classical transformations to reach IND-CPA then IND-CCA security levels. Note that, for example, from a high-level point of view, the key encapsulation mechanism BAT [FKPY22] follows essentially the same steps. The distinguishing problem we consider is relatively unexplored and consists in distinguishing a HNF of a co-cyclic lattice which is also diagonally dominant from a HNF of a random co-cyclic lattice.

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 $c = \texttt{DRE-Encrypt}(\mathbf{P}_K, \mathbf{m})$. See Definition 2 for the formal definition. Obviously, an adversary is allowed to produce as many pairs of plaintext-ciphertext as they want. 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 a matter of fact, one can prove that any advantage in the worst-case version of the one-wayness problem can be used to obtain an advantage in the recovery of the secret key. Indeed, any secret vector $\mathbf{B}_i = D \cdot \mathbf{e}_i + \mathbf{n}_i$ with $\|\mathbf{n}_i\|_1 < D$. For a noise levels $\nu(\mathbf{B})$ such as the ones chosen in our final choice of parameters (see §4.6), we get that with large probability, each coefficient of $\mathbf{n}_i$ belongs to $\{-1, 0, 1\}$ so $\mathbf{n}_i$ belongs to the message space $\mathcal{M}$. This implies that $\texttt{DRE-Encrypt}(\mathbf{P}_K, D \cdot \mathbf{e}_i) = \texttt{DRE-Encrypt}(\mathbf{P}_K, \mathbf{n}_i)$ so that recovering a secret vector can be turned into an instance of the one-wayness problem. Thus an adversary can turn an algorithm solving the one-wayness problem into an algorithm recovering the secret key. Note that this observation is backed up by the analysis of the concrete security done in Section 4.5.

It is classical to base formal security on distinguishing problems. In the case of our scheme, the path to go seems to be the problem of distinguishing co-cyclic diagonally dominant lattices from random ones. Indeed, given a perfect HNF from a random co-cyclic lattice and a perfect HNF from a random diagonally dominant matrix (or close like Goldreich-Goldwasser-Halevi (GGH)), we are currently unable to distinguish between the two without recovering the diagonally dominant base. The problem of distinguishing between random co-cyclic lattices and diagonally dominant lattices (that lie within the co-cyclic lattices) given their HNF does not seem to have been studied, but we argue that being able to distinguish lattices with diagonally dominant bases from a random co-cyclic lattice would break the system, and could finally break GGH after 20 years. Thus, let us describe how one could show that DRE is OW-CPA with more knowledge on diagonally dominant lattices. First, let us define the following distinguishing problem.

Definition 16 (Diagonally Dominant Distinguishing Problem ($\text{DIAGDOM}_{D,n,\mathcal{C}}$))**.** Consider $(D, n) \in \mathbb{N}$ and $\mathcal{C}$ a class of co-cyclic lattices with dimension n. Then the $\text{DIAGDOM}_{D,n,\mathcal{C}}$

problem asks on input a matrix H in HNF generated either from `DRE-KeyGen` or drawn uniformly from $\mathcal{C}$, to recover the right distribution.

In order to use this problem to build security proofs, one would like to know a class $\mathcal{C}$ such that $\text{DiagDom}_{D,n,\mathcal{C}}$ is hard. Since the determinant is an efficient sorting criterion, the class $\mathcal{C}$ should consist of lattices whose determinant lies in the range reached by `DRE-KeyGen`. Hopefully the determinants of diagonally dominant matrices belong to a *relatively* small interval I centered around D^n. Let us assume that this interval and the precise distribution $\mathcal{D}$ are known. Moreover, consider $\mathcal{C}$ to be the set of all co-cyclic lattices whose determinant follows $\mathcal{D}$. Now one has to remark that solving an instance of the OW-CPA game with DRE is reminiscent of the Group Inhomogeneous Short Integer Solution (GISIS) problem defined as follows.

Definition 17 ($\text{GISIS}_{n,\Delta,\beta}$)**.** Given a random co-cyclic lattice $\mathcal{L}$ of dimension n such that $\det \mathcal{L} = \Delta$ and a uniform $s \in \mathbb{Z}_\Delta$, find a vector $\mathbf{x}$ such that $\|\mathbf{x}\| \leqslant \beta$ and $\mathbf{x} \cdot \mathbf{H}[1..n, n] \equiv s \bmod \Delta$.

Following [GINX16], GISIS is believed to be hard and have been used to build secure cryptosystems, e.g. Squirrels [ENST23]. Combining the *conjectured* hardness of both $\text{DiagDom}_{D,n,\mathcal{C}}$ and GISIS, one can prove that DRE is OW-CPA. In order to take into account the fact that the determinant lies in an interval I with distribution $\mathcal{D}$ instead of being fixed, we introduce the $\text{GISIS}_{n,\mathcal{D},\beta}$ problem which consists in the average-case problem where Δ is drawn uniformly within I following $\mathcal{D}$.

Theorem 3. *For any adversary $\mathcal{A}$, there exists two adversaries of roughly the same running time of $\mathcal{A}$ such that* $\mathsf{Adv}^{\text{OW-CPA}}_{DRE}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{DiagDom}}_{D,n,\mathcal{C}}(\mathcal{A}_1) + \mathsf{Adv}^{\text{GISIS}}_{n,\mathcal{D},\sqrt{n}}(\mathcal{A}_2)$.

Proof. One can follow the proof of [FKPY22, Theorem 2]. Let us define two games.

- Game 1 is the standard OW-CPA game for DRE.
- Game 2 differs from Game 1 only in the key generation : Game 2 samples the public key in $\mathcal{C}$.

To build $\mathcal{A}_1$ for the $\text{DiagDom}_{D,n,\mathcal{C}}$ problem, we use this difference of distribution. On input H, $\mathcal{A}_1$ plays the challenger in Game 1 except setting $\mathbf{P}_K = H$ and claims that H is generated by `DRE-KeyGen` if $\mathcal{A}$ wins the game. Following [FKPY22, Theorem 2] we have that $\mathsf{Adv}^{\text{DiagDom}}_{D,n,\mathcal{C}}(\mathcal{A}_1) = |\mathbb{P}(W_1) - \mathbb{P}(W_2)|$, where W_i is the random event of $\mathcal{A}$ winning Game i, $i = 1, 2$. Then, on input a co-cyclic lattice $\mathcal{L}$ drawn in $\mathcal{C}$ and $s \in \mathbb{Z}_{\det \mathcal{L}}$, $\mathcal{A}_2$ plays the challenger in Game 2 except for setting $H = \text{HNF}(\mathcal{L})$. Then one sees that $\mathsf{Adv}^{\text{GISIS}}_{n,\mathcal{D},\sqrt{n}}(\mathcal{A}_2) = \mathbb{P}(W_2)$, which allows us to obtain the claimed inequality. □

IND-CPA from one-wayness. Assume that the scheme achieve OW-CPA security. Then, following [DHK+23, FKPY22] 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 modelised 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 [FO99, Den03].

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 $\mathbf{c}$ into $\mathbf{m}$ is not trivial without the secret key. Heuristically, if $\mathbf{c}$ is large enough, the problem of recovering $\mathbf{m}$ from $\mathbf{c}$ can be seen as a specific instance of the CVP, which is known to be hard. In practice, there is currently no better approach that solving this specific instance of CVP, which experimentally seems to behave exactly like over any other lattice [SPS19b, DY21], at least according to the estimates from [CN11, ADPS16]. With that in mind, what is left is the security of the public key. Since [Mic01], 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 perfect HNF 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 [PSDS18, SPS19a] or close structures [GGH97, MW01] are rather unsuccessful. The weaknesses were mostly on signature scheme instances [NR06, DN12, DY21] which do not concern this section. Note that [NR06] also consider that the *encryption* approach of [GGH97] is still secure, and to the extent of our knowledge this claim has not been challenged yet despite the public key *not being a HNF*.

4.5.1 Key recovery

Naive attack. The most naive attack is to reduce the public key using a BKZ-β algorithm 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 [DPPvW22], the required block size β 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 directly 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 $\mathbf{B}_i$ [5]. The cost of such an attack – without any additional knowledge – can be estimated following [ADPS16, AGVW17]. It is mentioned in [DY21] 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 the smallest block size β that BKZ can use to be potentially successful is fixed, one can apply the following conservative cost estimation for BKZ-β as used in [FKPY22]:

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

Note, that those "generic lattice estimates" were used by the authors of the original Diagonal Reduction Signature (DRS) submission [SPS19a] and the authors of the current attacks on DRS [DY21] do not seem to contradict the claims. In particular, [DY21]'s attack uses the estimates *after applying leakage knowledge* in order to estimate the efficiency of their attack. This of course does not constitute a proof that *no further attack exists*. As

[5] This method, named *Kannan's embedding* [Kan87], applies a reduction on a *specially extended* lattice of dimension $n+1$, similarly to the naive reduction on the original lattice of dimension n.

a matter of fact, it would be interesting to see if a new generic attack rises and answers the long-standing question: can we *finally* break the GGH encryption scheme concept in polynomial time? Thus, we have yet to see an attack that exploits the structure even further beyond the leak: as we stated earlier, the *encryption version* of GGH is still unchallenged[6], despite using an almost similar structure.

Attack on sparse keys. This previous attack can be eventually generalised 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 multiplied by the number of guesses required to succeed, which leads to:

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

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

4.5.2 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 $d(\mathbf{c}, \mathbf{v}) = \|\mathbf{m}\|$.

For key recovery, or at least recovery of one vector of the secret matrix[7], one needs to compute $\mathbf{N}_i$ from $D \cdot \mathbf{e}_i$ for any i, where $\mathbf{N}_i$ corresponds to a vector $\mathbf{N}_i \equiv \mathbf{B}_i - D \cdot \mathbf{e}_i \bmod \mathcal{L}(\mathbf{B})$. Clearly due to the sparsity of the non-diagonal elements in our system, it is easier to recover the very sparse noise vector $\mathbf{N}_i$ and thus $\mathbf{B}_i$ and the rest of the secret key, rather than a random message $\mathbf{m}$ which likely contains around a half of non-zero elements.

Therefore, the message recovery will be always more costly than the key recovery, if using the same lattice reduction technique. Since we are not aware of any attack that performs better on message recovery than key recovery, we consequently assume it will not be useful to evaluate message recovery security (as its cost will not be lower than the cost of key recovery attack).

4.6 Parameters and performance for DRE

4.6.1 Parametrisation philosophy

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 $12 \cdot \delta(\mathbf{B})$;

[6] At least after Nguyen's initial attack and fix [Ngu99].

[7] In practice, having one short vector helps recover the *other* short vectors of a short basis.

- the diagonal coefficient D is then deduced from $\delta(\mathbf{B})$ following the conditions described in Section 4.2.

To get an idea of the data sizes involved, it is important to notice that everything depends on the lattice volume, i.e its determinant $\det(\mathcal{L})$. Using the Hadamard bound, we can upper-bound the determinant by $\det(\mathcal{L}) \leqslant \log_2((\sqrt{D^2+\nu})^n)$, though in average we should have $\det(\mathcal{L}) \approx D^n$. Note that tighter bounds exist in the literature, but the Hadamard bound is easier to use.

- The public key has the size of $n\det(\mathcal{L})$.
- The ciphertext has the size of $\det(\mathcal{L})$.
- The secret key can be regenerated from the internal data of a `DRBG`, thus this is at least λ-bits. Storage of the noise matrix, which lies in $\{-1, 0, 1\}$, is also an option. In the latter case, the extreme sparsity of the matrix could also be leveraged to store the secret key, but we did not explore this direction. Note that the secret key needs to be reconstructed to perform the reduction algorithm, which itself is not fully optimised for the specific structure we have.

To fix the size of the precomputed data, we first need to fix a "granularity" α, in which we decompose the ciphertext of size $\det(\mathcal{L})$ by blocks of α-bits. Each block of the ciphertext is associated to a reduced vector of size $n\log_2(D)$ in the worst case. Thus, the size of the precomputed data can be measured in average to be $(\log_2(\det\mathcal{L})/\alpha)\cdot n\log_2(D)$ bits, which size entirely depends on α. Note, that we have arbitrarily chosen α to be 2^{16} in our experiments, leading to a precomputed data size to be higher than both secret and public key combined. It is not clear what would be the ideal parameters. While each precomputed vector can indeed have size lower than $n\log_2(D)$, we did not push the optimisation in this paper.

4.6.2 Benchmark comparison with FrodoKem

As a proof of concept, we implemented our encryption scheme in order to verify that it could obtain reasonable performances without aggressive optimisations. Indeed, since it uses lattices with no polynomial structure, one could wonder how slow a basic implementation of DRE would be. We provide timings (in number of cpucycles) from our basic implementation in `C`, and the associated parameters in Table 1. The tests were performed with a machine with an `Intel(R) Core(TM) i7-10710U CPU` with both hyperthreading and turboboost disabled. We also ran the benchmarks from the FrodoKem[BCD+16] package[8] for comparison. We also provide concrete numbers regarding the sizes of the objects we manipulate. The code is available online here:

https://codeberg.org/BaguetteInTodai/DiagonalDominantCryptography-CiC2025

The performance numbers of DRE are less than 10 times larger than the ones reported for the optimised implementation of FrodoKem, a third round NIST candidate. Furthermore, the code we tested lacks optimisation and is not feature complete: the code is mainly there to test whether or not the concept *algorithmically* works in practice and we did not implement the transformation from one-wayness to IND-CCA. Thus, it does not contain any additional security properties beyond the simple implementation of the algorithms. At this development stage, it is unclear whether it could be competitive with FrodoKem. Note that we did not list the timings of the public keys: we have inconsistent timings, relying on the `FLINT` library [9] to compute a HNF, with probabilistic rejection due to

[8] https://github.com/microsoft/PQCrypto-LWEKE/releases/tag/v1.0
[9] https://flintlib.org/

Table 1: Parameters for DRE and performance

Security level λ	128	192	256
DRE Dimension n	1920	2880	3840
DRE Noise level	160	240	320
DRE Diagonal D	483	723	963
DRE Encryption (avg cpucycles)	1092671	2358728	4456626
DRE Decryption (avg cpucycles)	18116442	43321208	79389958
FrodoKem Encryption (median cpucycles)	2416197	4546029	7736300
FrodoKem Decryption (median cpucycles)	2342903	4376348	7295394
DRE Max $\det(\mathcal{L})$ size (Hadamard, bytes)	2140	3420	4758
DRE Max $\mathbf{P}_K$ size (bytes)	4108800	9849600	18270720
FrodoKem $\mathbf{P}_K$ size (bytes)	9616	15632	21520

the fact that not every lattice sampled is co-cyclic: it can take up to several minutes to compute a single keypair. Optimisation is definitely possible: Squirrels's latest version also used a perfect HNF as a public key and reports "only" a few dozens of seconds, thus we could learn from their implementation if we plan to further optimise this new encryption cryptosystem.

Key sizes are definitely higher than those of FrodoKem. A tighter cryptanalysis or a different parametrisation method could maybe reduce the key sizes. Another path to improve data sizes and performance across the board may be to rely on another type of cryptographic construction using the decryption capacity of diagonal dominant matrices, such as a construction based on the *Lattice Isomorphism Problem (LIP)* [DvW22] or on compact gadgets [YJW23].

References

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim.

[AGVW17] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 297–322, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-70694-8_11.

[Ajt98] Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *30th Annual ACM Symposium on Theory of Computing*, pages 10–19, Dallas, TX, USA, May 23–26, 1998. ACM Press. doi:10.1145/276698.276705.

[AKKV05] Mikhail Alekhnovich, Subhash Khot, Guy Kindler, and Nisheeth K. Vishnoi. Hardness of approximating the closest vector problem with pre-processing. In *46th Annual Symposium on Foundations of Computer Science*, pages 216–225, Pittsburgh, PA, USA, October 23–25, 2005. IEEE Computer Society Press. doi:10.1109/SFCS.2005.40.

[Bab86] L. Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, Mar 1986. doi:10.1007/BF02579403.

[BBdV+17] Jens Bauch, Daniel J. Bernstein, Henry de Valence, Tanja Lange, and Christine van Vredendaal. Short generators without quantum computers: The case of multiquadratics. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 27–59, Paris, France, April 30 – May 4, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-56620-7_2.

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1006–1018, Vienna, Austria, October 24–28, 2016. ACM Press. doi:10.1145/2976749.2978425.

[BDK+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367, London, United Kingdom, April 24–26, 2018. IEEE Computer Society Press. doi:10.1109/EuroSP.2018.00032.

[BGPS23] Huck Bennett, Atul Ganju, Pura Peetathawatchai, and Noah Stephens-Davidowitz. Just how hard are rotations of $\mathbb{Z}^n$? algorithms and cryptography with the simplest lattice. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 252–281, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_9.

[BIP04] Jean-Claude Bajard, Laurent Imbert, and Thomas Plantard. Modular number systems: Beyond the Mersenne family. In Helena Handschuh and Anwar Hasan, editors, *SAC 2004: 11th Annual International Workshop on Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 159–169, Waterloo, Ontario, Canada, August 9–10, 2004. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-30564-4_11.

[BLNR22] Olivier Bernard, Andrea Lesavourey, Tuong-Huy Nguyen, and Adeline Roux-Langlois. Log-S-unit lattices using explicit stickelberger generators to solve approx ideal-SVP. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 677–708, Taipei, Taiwan, December 5–9, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-22969-5_23.

[BR20] Olivier Bernard and Adeline Roux-Langlois. Twisted-PHS: Using the product formula to solve approx-SVP in ideal lattices. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 349–380, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-64834-3_12.

[Bru82] Richard A Brualdi. Matrices eigenvalues, and directed graphs. *Linear and Multilinear Algebra*, 11(2):143–165, 1982. doi:10.1080/03081088208817439.

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585, Vienna, Austria, May 8–12, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-49896-5_20.

[CDW21] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Mildly Short Vectors in Cyclotomic Ideal Lattices in Quantum Polynomial Time. *J. ACM*, 68(2), January 2021. doi:10.1145/3431725.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20, Seoul, South Korea, December 4–8, 2011. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-25385-0_1.

[CPS82] JH Conway, RA Parker, and NJA Sloane. The Covering Radius of the Leech Lattice. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 261–290, 1982. doi:10.1098/rspa.1982.0042.

[dBS15] Charles F de Barros and L Menasché Schechter. GGH may not be dead after all. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 3(1), 2015. doi:10.5540/03.2015.003.01.0095.

[Den03] Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 133–151, Cirencester, UK, December 16–18, 2003. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-40974-8_12.

[DHK+23] Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, Gregor Seiler, and Dominique Unruh. A thorough treatment of highly-efficient NTRU instantiations. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 65–94, Atlanta, GA, USA, May 7–10, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-31368-4_3.

[Din00] Irit Dinur. Approximating SVP_∞ to within almost-polynomial factors is NP-hard. In Giancarlo Bongiovanni, Rossella Petreschi, and Giorgio Gambosi, editors, *Algorithms and Complexity*, pages 263–276, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. doi:10.1007/3-540-46521-9_22.

[DKL+21] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium algorithm specifications and supporting documentation. *Round-2 submission to the NIST PQC project*, 35, 2021. URL: https://pq-crystals.org/dilithium/data/dilithium-specification-round2.pdf.

[DLdW19] Emmanouil Doulgerakis, Thijs Laarhoven, and Benne de Weger. Finding closest lattice vectors using approximate voronoi cells. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 3–22, Chongqing, China, May 8–10, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-25510-7_1.

[DN12] Léo Ducas and Phong Q. Nguyen. Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 433–450, Beijing, China, December 2–6, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-34961-4_27.

[Dop14] Froilán M. Dopico. Diagonally dominant matrices: Surprising recent results on a classical type of matrices, 2014. URL: https://gauss.uc3m.es/fdopico/talks/2014-manch-nasc.pdf.

[DPPvW22] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel P. J. van Woerden. Hawk: Module LIP makes lattice signatures fast, compact and simple. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 65–94, Taipei, Taiwan, December 5–9, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-22972-5_3.

[dt23] The FPLLL development team. fpylll, a Python wrapper for the fplll lattice reduction library, Version: 0.6.0, 2023. Available at https://github.com/fplll/fpylll. URL: https://github.com/fplll/fpylll.

[DvW22] Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 643–673, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-07082-2_23.

[DY21] Léo Ducas and Yang Yu. Learning strikes again: The case of the DRS signature scheme. *Journal of Cryptology*, 34(1):1, January 2021. doi:10.1007/s00145-020-09366-9.

[ENST23] Thomas Espitau, Guilhem Niot, Chao Sun, and Medhi Tibouchi. Squirrels: Square Unstructured Integer Euclidean Lattice Signature, 2023. URL: https://squirrels-pqc.org.

[FKPY22] Pierre-Alain Fouque, Paul Kirchner, Thomas Pornin, and Yang Yu. BAT: Small and fast KEM over NTRU lattices. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(2):240–265, 2022. doi:10.46586/tches.v2022.i2.240-265.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-48405-1_34.

[Ger31] Semyon Aronovich Gerschgorin. 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. URL: https://www.mathnet.ru/eng/im5235.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski, Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*,

pages 112–131, Santa Barbara, CA, USA, August 17–21, 1997. Springer Berlin Heidelberg, Germany. doi:10.1007/BFb0052231.

[GINX16] Nicolas Gama, Malika Izabachène, Phong Q. Nguyen, and Xiang Xie. Structural lattice reduction: Generalized worst-case to average-case reductions and homomorphic cryptosystems. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 528–558, Vienna, Austria, May 8–12, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-49896-5_19.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press. doi:10.1145/1374376.1374407.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, June 1998. doi:10.1007/BFb0054868.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987. doi:10.1287/moor.12.3.415.

[Kle00] Philip N. Klein. Finding the closest lattice vector when it's unusually close. In David B. Shmoys, editor, *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 937–941, San Francisco, CA, USA, January 9–11, 2000. ACM-SIAM. doi:10.5555/338219.338661.

[Lim82] David JN Limebeer. The application of generalized diagonal dominance to linear system stability theory. *International Journal of Control*, 36(2):185–212, 1982. doi:10.1080/00207178208932886.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. doi:10.1007/BF01457454.

[LPS20] Andrea Lesavourey, Thomas Plantard, and Willy Susilo. Short principal ideal problem in multicubic fields. *Journal of Mathematical Cryptology*, 14(1):359–392, 2020. URL: https://doi.org/10.1515/jmc-2019-0028 [cited 2025-06-22], doi:doi:10.1515/jmc-2019-0028.

[LSZ+24] Xiuhan Lin, Moeto Suzuki, Shiduo Zhang, Thomas Espitau, Yang Yu, Mehdi Tibouchi, and Masayuki Abe. Cryptanalysis of the Peregrine lattice-based signature scheme. In Qiang Tang and Vanessa Teague, editors, *PKC 2024: 27th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 14601 of *Lecture Notes in Computer Science*, pages 387–412, Sydney, NSW, Australia, April 15–17, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-57718-5_13.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. The deep space network progress report 42-44, Jet Propulsion Laboratory, California Institute of Technology, January/February 1978. https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[MG12] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2012. doi:10.1007/978-1-4615-0897-7.

[MH78] Ralph Merkle and Martin Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE transactions on Information Theory*, 24(5):525–530, 1978. doi:10.1109/TIT.1978.1055927.

[Mic01] Daniele Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *International Cryptography and Lattices Conference*, pages 126–145. Springer, 2001. doi:10.1007/3-540-44670-2_11.

[Min96] Hermann Minkowski. *Geometrie der Zahlen*. B.G. Teubner, Leipzig, 1896.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009. doi:10.1007/978-3-540-88702-7_5.

[MW01] Daniele Micciancio and Bogdan Warinschi. A linear space algorithm for computing the Hermite normal form. In *Proceedings of the 2001 international symposium on Symbolic and algebraic computation*, pages 231–236. ACM, 2001. doi:10.1145/384101.384133.

[Ngu99] Phong Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto'97. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 288–304, Santa Barbara, CA, USA, August 15–19, 1999. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-48405-1_18.

[NIS18] NIST. Post-quantum cryptography standardization, 2018. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography.

[NIS23a] NIST. Post-Quantum Cryptography: Digital Signature Schemes, 2023. URL: https://csrc.nist.gov/projects/pqc-dig-sig/standardization/call-for-proposals.

[NIS23b] NIST. Post-Quantum Cryptography Standardization, 2023. URL: https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 271–288, St. Petersburg, Russia, May 28 – June 1, 2006. Springer Berlin Heidelberg, Germany. doi:10.1007/11761679_17.

[NS16] Phong Q. Nguyen and Igor E. Shparlinski. Counting co-cyclic lattices. *SIAM Journal on Discrete Mathematics*, 30(3):1358–1370, 2016. doi:10.1137/15M103950X.

[PHS19] Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. Approx-SVP in ideal lattices with pre-processing. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 685–716, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-17656-3_24.

[PSDS18] Thomas Plantard, Arnaud Sipasseuth, Cédric Dumondelle, and Willy Susilo. DRS : Diagonal dominant reduction for lattice-based signature. PQC Standardization Conference, Round 1 submissions, 2018. URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/DRS.zip.

[PSW08] Thomas Plantard, Willy Susilo, and Khin Than Win. A digital signature scheme based on CVP$_{infinity}$. In Ronald Cramer, editor, *PKC 2008: 11th International Workshop on Theory and Practice in Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 288–307, Barcelona, Spain, March 9–12, 2008. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-78440-1_17.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *35th Annual ACM Symposium on Theory of Computing*, pages 407–416, San Diego, CA, USA, June 9–11, 2003. ACM Press. doi:10.1145/780542.780603.

[Rum18] Siegfried M Rump. Estimates of the determinant of a perturbed identity matrix. *Linear algebra and its applications*, 558:101–107, 2018. doi:10.1016/j.laa.2018.08.009.

[SKLJS22] Eun-Young Seo, Young-Sik Kim, Joon-Woo Lee, and No Jong-Seon. Peregrine: Submission to the Korea post-quantum cryptography competition, 2022. URL: https://www.kpqc.or.kr/competition.html.

[SPS19a] Arnaud Sipasseuth, Thomas Plantard, and Willy Susilo. Enhancing Goldreich, Goldwasser and Halevi's scheme with intersecting lattices. *Journal of Mathematical Cryptology*, 13(3-4):169–196, 2019. doi:10.1515/jmc-2016-0066.

[SPS19b] Arnaud Sipasseuth, Thomas Plantard, and Willy Susilo. Improving the security of the DRS scheme with uniformly chosen random noise. In Julian Jang-Jaccard and Fuchun Guo, editors, *ACISP 19: 24th Australasian Conference on Information Security and Privacy*, volume 11547 of *Lecture Notes in Computer Science*, pages 119–137, Christchurch, New Zealand, July 3–5, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-21548-4_7.

[SPS20] Arnaud Sipasseuth, Thomas Plantard, and Willy Susilo. A noise study of the PSW signature family: Patching DRS with uniform distribution. *Information*, 11(3), 2020. URL: https://www.mdpi.com/2078-2489/11/3/133, doi:10.3390/info11030133.

[Tau49] Olga Taussky. A recurring theorem on determinants. *The American Mathematical Monthly*, 56(10P1):672–676, 1949. doi:10.1080/00029890.1949.11990209.

[YD18] Yang Yu and Léo Ducas. Learning strikes again: The case of the DRS signature scheme. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 525–543, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-030-03329-3_18.

[YJW23] Yang Yu, Huiwen Jia, and Xiaoyun Wang. Compact lattice gadget and its applications to hash-and-sign signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-38554-4_13.

A Reparametrisation of DRS

We mentioned earlier that the attack on DRS [DY21] has exponential complexity. Thus in theory, it is possible to reparametrise the original DRS scheme to heuristically restore security levels through polynomial increase of the mathematical objects used in the scheme. In this part of the appendix, we provide such a reparametrisation for the most popular NIST security levels: 128, 192 and 256-bits. We re-use the building blocks of DRE to achieve such a reparametrisation.

A.1 Quick recap of the DRS scheme and attacks

The signature scheme called DRS was a submission to the first round of the NIST standardisation process for quantum resistant signature schemes [NIS18], using diagonally dominant lattices. The main idea of DRS is to follow a framework close the one of GGH [GGH97] but using the diagonal dominance property to sign within a hypercube independent of the secret key, hoping to prevent leaking the secret key as in [NR06] for example. This was first presented by Plantard et al. in [PSW08]. However the original DRS scheme has been subject to a learning attack from Ducas and Yu [YD18], which was then extended to the second version of the scheme, the so-called DRSv2 [SPS20]) [DY21].

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 guesses $\mathbf{B}'_i$ 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.

A.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 3 we plotted the factors $r(n, N)$ that we obtained using our new parameters for the DRS scheme.

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 stabilise 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 [DY21] should not have an important impact on the asymptotic security of the DRS scheme with our new parameters.

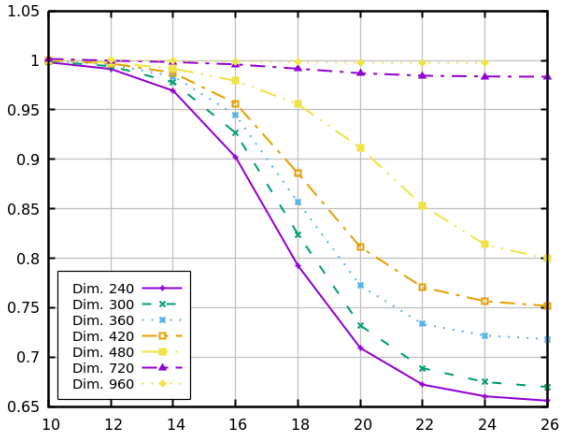

Figure 3: New experimental measures of $r(n, N)$.

A.3 Security analysis

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 4.5.1 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 [NR06] and its improvements [DN12, LSZ+24] 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 [NR06], 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. [DY21] 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

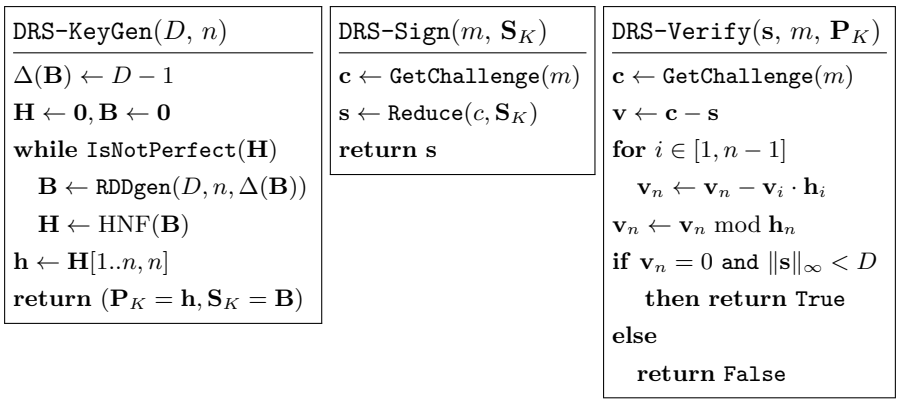

Figure 4: DRS Algorithms as DRE variant

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 leads us to consider the setting of the Closest Vector Problem with Preprocessing (CVPP).

Assume that a learning attack *à la* Nguyen and Regev [NR06, DN12, LSZ+24] allows us to recover vectors from a hidden parallelotope 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 Section C that the average approximation factor reached by Algorithm 1, both for signed or negative noises, is a small constant. Following [DLdW19] 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 computing the shortest vectors (up to some approximation factors) of the lattice, among which are the vectors of the secret basis. Thus, one would certainly recover the secret basis as a by-product of the query phase. Thus, we deem that forging a signature using (approximate) Voronoi cells or classical algorithms solving the CVPP_γ [DLdW19] is as hard as recovering the secret key.

A.4 DRS as a variant of DRE

A.4.1 Differences with the original DRS description

In figure 4 we describe how DRS can be described as a signing version of DRE, similarly to how GGHSign is a variant of GGHEncrypt. The differences with the original description of DRS in [PSDS18, SPS19b] lie mostly in a higher abstraction level and the use of a HNF, and we do not describe precomputations here as they do not necessarily provide any advantage. To simplify the description we use a deterministic `GetChallenge` function that transforms the message into a target vector to reduce. This is a typical hash-then-sign approach.

A.4.2 Choice of the challenge generation

We present two ways to generate the challenge through `GetChallenge`, i.e. how to transform the message into an instance of the GDD_γ so that the signatory (or the forger) is ablet to produce a valid signature.

1. We can opt for choosing a random integer modulo the determinant, which has similar form as an *encryption* of a ciphertext in DRE, i.e

$$\texttt{GetChallenge}(m) \leftarrow \mathbf{c} = [0, ..., 0, c].$$

   The signing process then reduces this vector consisting of only one non-zero (but large) integer and the verification can be simplified by skipping the subtraction $\mathbf{v} - \mathbf{c}$ and then checking $\mathbf{v}_n = c$ at the end of the modular scalar product (which then uses smaller entries). We can use precomputations for the signing process just like in the DRE decryption process, but this would still yield a much slower signature for only a minor increase in verification speed.

2. We can opt for the classic DRS procedure, i.e generate a random vector of large entries, i.e

$$\texttt{GetChallenge}(m) \leftarrow \mathbf{c} \text{ with } c \in [-(2^{2\log_2(D)}), 2^{2\log_2(D)}]^n$$

   then reduce this large vector. This produces a faster signature than the above method, but verification requires reducing the difference of the vector through the public key HNF and check that the resulting reduction is the zero vector, which is a bit more costly than reducing a small vector. n this context, precomputations do not make sense since the maximum norm of the challenge vector is already quite small.

Both generation methods use the message as a seed for an expander, either through hashing (`SHAKE` for example) or using a stream cipher in counter mode (`AES256-CTR` for example).

A.5 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	192	256
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 submitted as a candidate for the NIST PQC standardisation by Squirrels [ENST23], while our verification is less than 6 times slower. While the code is suboptimal and very basic (and implemented using the *first* option for challenge generation using DRE functions), we believe it performs well enough to claim the approach is somehow still reasonable to be studied.

B 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 centred 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 longer depend on $\Delta(\mathbf{B})$ anymore. This could be used when diagonally dominant matrices are used for cryptography.

C Average quality of reduction

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 [dt23], 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 [AKKV05].

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

	$\Delta(\mathbf{B})$	dimension n							
		10	20	30	40	45	50	55	60
`PSW`	1	2.91	2.79	2.67	2.61	2.56	2.56	2.51	2.50
	$D/2$	1.55	1.50	1.38	1.43	1.36	1.38	1.36	1.38
`neg-PSW`	1	1.44	1.26	1.22	1.18	1.15	1.15	1.13	1.14
	$D/2$	1.066	1.028	1.021	1.018	1.010	1.011	1.010	1.009

D Short vectors and reduction algorithms for Column Diagonally Dominant matrices

In this section we consider Column Diagonally Dominant matrices. A c.d.d. matrix can be simply defined as the transpose matrix of a diagonally dominant matrix on the rows

(definition 15). Consequently, we will write $\Delta^T(\mathbf{B}) = \Delta(\mathbf{B}^T)$.

The overall methodology used in this section is very similar to the one of Section 3. Again, the results proven in this section can be grouped in the following theorem.

Theorem 4. *Consider $\mathbf{B} \in \mathbb{Z}^n$ a c.d.d. matrix and $\mathcal{L} = \mathcal{L}(\mathbf{B})$. Then $\lambda_1^{(\infty)}(\mathcal{L}) \geqslant \Delta^T(\mathbf{B})$ and there is an algorithm, `RSR` (Alg. 3), running within a polynomial amount of arithmetic operations such that*

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

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`.

D.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 $\mathcal{B}_I \neq \lambda_1(\mathbf{B})$.

D.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

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

□

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

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

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

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

The reduction algorithm Algorithm 1 only concerned r.d.d matrices and is 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 $q = \max\{t \in \mathbb{N}^* \mid \forall i \in [\![1,n]\!], \|\mathbf{v}\|_\infty - tR_i \geqslant t(\delta_i(\mathbf{B}^T))\}$

$w \leftarrow v$, $i \leftarrow 1$, $s \leftarrow [0, ..., 0] \in \{0,1\}^n$ {initialisation 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 an 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

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

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

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

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: $w \leftarrow$`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 Babai's nearest plane algorithm.