University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part B

Faculty of Engineering and Information
Sciences

2019

Improving the Security of the DRS Scheme with Uniformly Chosen Random Noise

Arnaud Sipasseuth

University of Wollongong, as447@uowmail.edu.au

Thomas Plantard

University of Wollongong, thomaspl@uow.edu.au

Willy Susilo

University of Wollongong, wsusilo@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers1>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Improving the Security of the DRS Scheme with Uniformly Chosen Random Noise

Abstract

At PKC 2008, Plantard et al. published a theoretical framework for a lattice-based signature scheme. Recently, after ten years, a new signature scheme dubbed as the Diagonal Reduction Signature (DRS) scheme was presented in the NIST PQC Standardization as a concrete instantiation of the initial work. Unfortunately, the initial submission was challenged by Yu and Ducas using the structure that is present on the secret key noise. In this paper, we are proposing a new method to generate random noise in the DRS scheme to eliminate the aforementioned attack, and all subsequent potential variants.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Sipasseuth, A., Plantard, T. & Susilo, W. (2019). Improving the Security of the DRS Scheme with Uniformly Chosen Random Noise. Lecture Notes in Computer Science, 11547 LNCS 119-137.

Improving the security of the DRS scheme with uniformly chosen random noise

Arnaud Sipasseuth^{1,2}, Thomas Plantard^{1,3}, and Willy Susilo^{1,4}

¹ *iC²*: Institute of Cybersecurity and Cryptology
University of Wollongong,
School of Computing and Information Technology

² `as447@uowmail.edu.au`

³ `thomaspl@uowmail.edu.au`

⁴ `wsusilo@uow.edu.au`

Abstract. At PKC 2008, Plantard et al. published a theoretical framework for a lattice-based signature scheme. Recently, after ten years, a new signature scheme dubbed as the Diagonal Reduction Signature (DRS) scheme was presented in the NIST PQC Standardization as a concrete instantiation of the initial work. Unfortunately, the initial submission was challenged by Yu and Ducas using the structure that is present on the secret key noise. In this paper, we are proposing a new method to generate random noise in the DRS scheme to eliminate the aforementioned attack, and all subsequent potential variants.

Keywords: Lattice-based cryptography, DRS, Lattice-based signatures, NIST PQC, Diagonal Dominant

1 Introduction

The popularity of post-quantum cryptography has increased significantly after the formal announcement by the National Institute of Standards and Technology (NIST) to move away from classical cryptography [18]. This is due to the potential threat that will be brought by the upcoming large scale quantum computers, which theoretically break the underlying traditional hard problem by using Shor’s algorithm [25]. There are currently three main families in post-quantum cryptology, namely code-based cryptography, multivariate cryptography, and lattice-based cryptography. This work primarily concerns with lattice-based cryptography. First introduced by Minkowski in a pioneering work [15] to solve various number problems, lattices have the advantage to often base their security on worst-case assumptions [1] rather than the average case, and to be highly parallelizable and algorithmically simple enough to compete with traditional schemes in terms of computing speed. Inspired by this, Goldreich, Goldwasser and Halevi (GGH) [7] proposed an efficient way to use lattices to build a public-key encryption scheme. Their practical scheme has been broken using lattice reduction techniques [16], however the central idea remains viable and it has enabled a wide array of applications and improvements, such as using tensor products [5], Hermite Normal Forms [14], polynomial representations [19],

rotations [26], and the most popular one being Learning With Errors [23] or its variants.

More recently, the NIST attempt at standardizing post-quantum cryptography [17] received a lot of interest from the community and the vast majority of the lattice-based submissions for “Round 1” are actually based on LWE [17]. One of the few lattice-based submissions which is not using LWE or ideal lattices is the Diagonal Reduction Signature Scheme (DRS) [20], which uses a diagonal dominant matrix that can be seen as a sum between a diagonal matrix with very big coefficients and a random matrix with low coefficients. DRS was based on a paper from PKC 2008 [21] however the original paper had mostly a theoretical interest and did not provide an explicit way to construct the random matrix with low values, rather than merely stating conditions on norm bounds it should respect for the signature scheme to be proven functioning. The NIST submission however provides a more straight-forward way to generate the noise, using another proof and condition to ensure the functionality of the scheme. This new way to generate the noise, however, is shown to be insecure: soon after DRS was made public, Yu and Ducas used machine learning techniques to severely reduce the security parameters [30]. Although according to Ducas’ comments on the NIST forum [17], the attack was not devastating as it still seems asymptotically secure, however its concrete security was significantly decreased. On the same work, Yu and Ducas also provided several suggestions in order to fix those issues and one of those comments suggested using a statistical analysis. Another more recent attack from Li, Liu, Nitaj and Pan [11] on a randomized version of the initial scheme proposed by Plantard, Susilo and Win [21] can also be indirectly considered an attack to the DRS scheme, although this attack does not seem as important as Yu and Ducas’s one.

In the following work, we do follow some of those suggestions and we aim to provide a new noise generation method to eliminate the aforementioned attack and restore some of the DRS’ concrete security. We will present some statistical heuristics and remove some of the structure that allow the initial DRS scheme to be attacked.

Our Contribution and Paper Organization

The rest of the paper is organized as follows. We first present some relevant background on lattice theory and re-introduce the DRS scheme from Plantard et al. Subsequently, we will comment on the attack of Li, Liu, Nitaj and Pan [11] and explain why it is not applicable. Then we discuss the weakness found by Yu and Ducas and our idea to correct this. We finally present the detail algorithms about our security patch and raise some open questions.

2 Background

In this section, we briefly recall the basics of lattice theory.

2.1 Lattice theory

Definition 1 *We call lattice a discrete subgroup of $\mathbb{R}^n$ where n is a positive integer. We say a lattice is an integer lattice when it is a subgroup of $\mathbb{Z}^n$. A*

basis of the lattice is a basis as a $\mathbb{Z}$ -module. If M is a matrix, we define $\mathcal{L}(M)$ the lattice generated by the rows of M .

In this work we only consider full-rank integer lattices, i.e., such that their basis can be represented by a $n \times n$ non-singular integer matrix.

Theorem 1 (Determinant) *For any lattice $\mathcal{L}$, there exists a real value we call **determinant**, denoted $\det(\mathcal{L})$, such that for any basis B , $\det(\mathcal{L}) = \sqrt{\det(BB^T)}$.*

The literature sometimes call $\det(\mathcal{L})$ as the **volume** of $\mathcal{L}$ [15].

Definition 2 *We say a lattice is a diagonally dominant type lattice if it admits a basis of the form $D + R$ where $D = d \times Id$, $d \in \mathbb{Z}$, and R is a “noise” matrix whose diagonal entries are zeroes and the absolute sum of each entry is lower than d per each row separately.*

We note that the definition is similar to the one which can be found in fundamental mathematics books [3] for diagonal dominant matrices. We will just adapt the lattice to its diagonal dominant basis.

Definition 3 (Minima) *We note $\lambda_i(\mathcal{L})$ the i -th minimum of a lattice $\mathcal{L}$. It is the radius of the smallest zero-centered ball containing at least i linearly independent elements of $\mathcal{L}$.*

Definition 4 (Lattice gap) *We note $\delta_i(\mathcal{L})$ the ratio $\frac{\lambda_{i+1}(\mathcal{L})}{\lambda_i(\mathcal{L})}$ and call that a lattice gap. When mentioned without index and called “the” gap, the index is implied to be $i = 1$.*

In practice, only the case $i = 1$ is used, but other values are sometimes useful to consider [29]. We also define the “root lattice gap”, i.e., elevated to the power $\frac{1}{n}$ where n is the dimension of the lattice.

Definition 5 *We say a lattice is a diagonally dominant type lattice (of dimension n) if it admits a diagonal dominant matrix as a basis B as in [3], i.e.,*

$$\forall i \in [1, n], B_{i,i} \geq \sum_{j=1, i \neq j}^n |B_{i,j}|$$

We can also see a diagonally dominant matrix B as a sum $B = D + R$ where D is diagonal and $D_{i,i} > \|R_i\|_1$. In our scheme, we use a diagonal dominant lattice as our secret key, and will refer to it as our “reduction matrix” (as we use this basis to “reduce” our vectors).

Definition 6 *Let F be a subfield of $\mathbb{C}$, V a vector space over F^k , and p a positive integer or ∞ . We call l_p norm over V the norm:*

$$\begin{aligned} - \forall x \in V, \|x\|_p &= \sqrt[p]{\sum_{i=1}^k |x_i|^p} \\ - \forall x \in V, \|x\|_\infty &= \max_{i \in [1, k]} |x_i| \end{aligned}$$

l_1 and l_2 are commonly used and are often called taxicab norm and euclidean norm, respectively. We note that we also define the maximum matrix norm as the biggest value among the sums of the absolute values in a single column.

The norm that was used by Plantard et al. for their signature validity is the maximum norm. However, as far as the security heuristics are concerned the euclidean norm (l_2) is used, and as far as the reduction termination proof is concerned the taxicab norm (l_1) is used.

2.2 Lattice problems

The most famous problems on lattice are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). We tend to approximatively solving CVP by solving heuristically SVP in an expanded lattice [7].

Definition 7 (CVP: Closest Vector Problem) *Given a basis B of a lattice $\mathcal{L}$ of dimension n and $t \in \mathbb{R}^n$, find $v \in \mathcal{L}$ such that $\forall w \in \mathcal{L}, \|t - v\| \leq \|t - w\|$.*

Definition 8 (SVP: Shortest Vector Problem) *Given a basis B of a lattice $\mathcal{L}$ of dimension n , find $v \in \mathcal{L}$ such that $\forall w \in \mathcal{L}, v \neq w, \|v\| \leq \|w - v\|$, i.e. $\|v\| = \lambda_1(B)$.*

In cryptography, we rely on the “easier” versions of those problems:

Definition 9 (uSVP $_\delta$: δ -unique Shortest Vector Problem) *Given a basis of a lattice $\mathcal{L}$ with its lattice gap $\delta > 1$, solve SVP.*

Since $\lambda_1(\mathcal{L})$ is also hard to determine (it is indeed another lattice problem we do not state here), measuring the efficiency of an algorithm is another challenge by itself. Therefore, to measure algorithm efficiency we must be able to define a problem with easily computable parameters, which is where the Hermite factor is originated from:

Definition 10 (HSVP $_\gamma$: γ -Hermite Shortest Vector Problem) *Given a basis B of a lattice $\mathcal{L}$ of dimension n and a factor γ we call Hermite Factor, find $y \in \mathcal{L}$ such that $\|y\| \leq \gamma \det(\mathcal{L})^{1/n}$.*

Some cryptosystems are based on worst-case hardness on uSVP with polynomial gap as [2] and [23]. The practical hardness of uSVP depends on its gap compared to a fraction of the Hermite factor, where the constant in front of the factor depends of the lattice and the algorithm used [6]. There exists an attack that was specifically built to exploit high gaps [12].

Definition 11 (BDD $_\gamma$: γ -Bounded Distance Decoding) *Given a basis B of a lattice $\mathcal{L}$, a point x and a approximation factor γ ensuring $d(x, \mathcal{L}) < \gamma \lambda_1(B)$ find the lattice vector $v \in \mathcal{L}$ closest to x .*

It has been proved that **BDD** $_{1/(2\gamma)}$ reduces itself to **uSVP** $_\gamma$ in polynomial time and the same goes from **uSVP** $_\gamma$ to **BDD** $_{1/\gamma}$ when γ is polynomially bounded by n [13], in cryptography the gap is polynomial the target point x must be polynomially bounded therefore solving one or the other is relatively the same in our case. To solve those problems, we usually use an embedding technique that extends a basis matrix by one column and one row vector that are full of zeroes except for one position where the value is set to 1 at the intersection of those newly added spaces, and then apply lattice reduction techniques on these. As far as their signature scheme is concerned, the **GDD** $_\gamma$ is more relevant:

Definition 12 (GDD $_\gamma$: γ -Guaranteed Distance Decoding) *Given a basis B of a lattice $\mathcal{L}$, any point x and a approximation factor γ , find $v \in \mathcal{L}$ such that $\|x - v\| \leq \gamma$.*

3 The initial DRS scheme and its security pitfall

We will briefly summarize the DRS scheme below, which can be considered a fork of the theoretical framework of PKC 2008 [21]. The DRS scheme uses the maximum norm to check if a vector is reduced. To achieve that purpose, they use a diagonal dominant basis, where every subtraction from a diagonally dominant basis vector reduces a coefficient by a lot more than it potentially adds to the other coefficients. By repeating those steps for each coefficient, we end up reducing the vector. The initial DRS scheme requires multiple parameters to be preset (see the file *api.h* in their NIST submission), which we give here the main ones describing their choice for a secret key: D , a big diagonal coefficient, N_B , the number of occurrences per vector of the “big” noise $\{-B, B\}$, and is the lowest positive number such that $2^{N_B} \binom{n}{N_b} \geq 2^\lambda$, B , the value of the “big” noise, and is equal to $D/(2N_B)$, and N_1 , the number of occurrences per vector of the small noise $\{-1, 1\}$, and is equal to $D - (N_B B) - \Delta$. As we will see by discussing previous work, this structure directly impact the security.

3.1 The original DRS scheme

Setup Using the same notation as the report given in [20], we briefly restate their initial algorithm. Those parameters are chosen such that the secret key matrix stays diagonal dominant as per the definition written previously. From our understanding, the large coefficients B were used to increase the euclidean norm, as an attempt to enhance its security against lattice reduction attacks. Alg 1 is the original secret key computation.

The public key is obtained by successive additions/subtractions of pair of vectors (see Alg 2). Note that the only difference with the original scheme is that we do not store the $\log_2$ of the maximum norm. We estimate this information to be easily computed at will.

This is equivalent to a multiplication of random pairs of vectors (a $2 \times n$ matrix) by a square invertible matrix of dimension 2 and maximum norm of 2. In their case, every vector go through exactly one matrix multiplication per round, for a total of R rounds where R is defined by the system. The number of rounds R is decided upon security consideration but also efficiency reasons as the authors of DRS wanted to fit every computation within 64-bits. For more details we refer again to [20]. From our understanding, the power of 2 p_2 has no security impact, and is used mostly for the verification process to make sure intermediate computation results stay within 64-bits. This type of public key is very different from the Hermite Normal Form proposed in [21], however the computation time of a Hermite Normal Form is non-negligible. As we will see later this directly impact the signature.

Signature Given the fact that the secret key is a diagonally dominant matrix, Alg 3 is guaranteed to complete. The proof can be seen in [20]. Plantard et al.

```

Input: - all initial parameters;
- another extra random seed  $x_2$ ;
Output: -  $x, S$  the secret key;
// Initialization
 $S \leftarrow 0$ ;
 $t \in \mathbb{Z}^n$ ;
// Algorithm start
InitiateRdmSeed( $x_2$ );
// Set  $t[1]$  to  $D$ ,  $N_B$  elements to  $B$ ,  $N_1$  to 1, the rest to 0
 $t \leftarrow [D, \underbrace{B, \dots, B}_{N_B}, \underbrace{1, \dots, 1}_{N_1}, 0, \dots, 0]$ ;

// Randomly permute values of  $t$  with a function RdmPmtn
// RdmPmtn leaves  $t[1]$  unchanged
 $t \leftarrow \text{RdmPmtn}(t)$ ;
for  $i = 1 ; i \leq n ; i = i + 1$  do
     $S[i][i] \leftarrow t[1]$ ;
    // Apply a circular permutation and randomly flip signs
    for  $j = 2 ; j \leq n ; j = j + 1$  do
         $S[i][((i + j) \bmod n) + 1] \leftarrow t[j] * \text{RdnSgn}()$ ;
    end
end
return  $x, S$ ;

```

Algorithm 1: Secret key generation (square matrix of dimension n)

```

Input: -  $S$  the reduction matrix of dimension  $n$ , obtained previously;
- a random seed  $x$ ;
Output: -  $P$  the public key, and  $p_2$  a power of two;
// Initialization
 $P \leftarrow S$ ;
// Algorithm start
InitiateRdmSeed( $x$ );
// Apply  $R$  rounds
for  $i = 1 ; i < R ; i = i + 1$  do
     $P \leftarrow \text{RdmPmtn}(P)$ ;
    for  $j = 1 ; j \leq n - 1 ; j = j + 2$  do
         $t \leftarrow \text{RdmSgn}()$ ;
         $P[j] = P[j] + t * P[j + 1]$ ;
         $P[j + 1] = P[j + 1] + t * P[j]$ ;
    end
end
 $P \leftarrow \text{RdmPmtn}(P)$ ;
return  $P, p_2$ ;

```

Algorithm 2: Public key generation

did not have a second vector k to output in their initial scheme and thus only had to deal with the reduction part [21].

```

Input: - A vector  $v \in \mathbb{Z}^n$ ;
-  $S$  the secret key matrix, with diagonal coefficient  $d$ ;
-  $s$  a seed value;
Output: -  $w$  with  $v \equiv w \pmod{S}$ ,  $\|w\|_\infty < d$  and  $k$  with  $kP = v - w$ ;
// Initialization
 $w \leftarrow v$ ,  $i \leftarrow 0$ ,  $k \leftarrow [0, \dots, 0]$ ;
// Algorithm start
// Reduce until all coefficients are low enough
while  $\|w\|_\infty < d$  do
     $q \leftarrow \frac{w_i}{d}$ ,  $k_i \leftarrow k_i + q$ ,  $w \leftarrow w - qS[i]$ ,  $i \leftarrow i + 1 \bmod n$ ;
end
// Use the seed to modify  $k$  such that  $kP = v - w$ 
// The seed defines the output of RdmPmtn and RdmSgn
InitiateRdmSeed( $s$ );
for  $i = 1$  ;  $i \leq R$  ;  $i = i + 1$  do
     $k \leftarrow \text{RdmPmtn}(k)$ ;
    for  $j = 1$  ;  $j \leq n - 1$  ;  $j = j + 2$  do
         $t \leftarrow \text{RdmSgn}()$ ;
         $k[j + 1] = k[j + 1] - t * k[j]$ ;
         $k[j] = k[j] - t * k[j + 1]$ ;
    end
end
 $k \leftarrow \text{RdmPmtn}(k)$ ;
return  $k, v, w$ ;

```

Algorithm 3: Sign : coefficient reduction first, validity vector then

Verification Alg 4 checks if the vector v is reduced enough i.e $\|v\|_\infty < D$ where D is the diagonal coefficient of the secret key matrix S . Then it tries to check the validity of $kP = v - w$. By using the power p_2 , the authors of DRS want to ensure the computations stay within 64-bits. If multiprecision integers were used (as GMP), we note it would not take a while loop with multiple rounds to check. Whether this is more efficient or not remains to be tested.

3.2 Li, Liu, Nitaj and Pan's attack on a randomized version of the initial PKC'08

In ACISP 2018, Li, Liu, Nitaj and Pan [11] presented an attack that makes use of short signatures to recover the secret key. Their observation is that two different signatures from the same message is also a short vector of the lattice. Then, gathering sufficient number of short vectors enable easier recovery of the

```

Input: - A vector  $v \in \mathbb{Z}^n$ ;
-  $P, p_2$  the public key matrix and the  $\log_2$  of its maximum norm;
-  $w$  the reduced form of  $v$ ;
-  $k$  the extra information vector;
Output: -  $w$  a reduced vector, with  $v \equiv w [\mathcal{L}(P)]$ ;
// Test for max norm first
if  $\|w\|_\infty \geq D$  then return FALSE;
// Loop Initialization
 $q \leftarrow k, t \leftarrow v - w$ ;
while  $q \neq 0 \wedge t \neq 0$  do
     $r \leftarrow q \bmod p_2, t \leftarrow rP - t$ ;
    // Check correctness
    if  $t \neq 0 \bmod y$  then return FALSE;
     $t \leftarrow t/p_2, q \leftarrow (q - r)/p_2$ ;
    if  $(t = 0) \vee (q = 0)$  then return FALSE;
end
return TRUE;

```

Algorithm 4: Verify

secret key using lattice reduction algorithms with the vectors generated. Their suggestion to fix this issue is to either store previous signed messages to avoid having different signatures, or padding a random noise in the hash function. We should note that the initial DRS scheme is not randomized as the algorithm is deterministic and produce a unique signature per vector.

We do note that the authors of DRS suggested in their report [20] to use a random permutation to decide the order of the coefficient reduction, and thus Li, Liu, Nitaj and Pan's attack might apply to their suggestion. However, the order of the coefficient reduction could also be decided deterministically by the hashed message itself, and therefore, Li, Liu, Nitaj and Pan's attack is not fully applicable, as this method would produce an unique signature per message. They can still generate a set of relatively short vectors $(r_1, \dots, r_2) \in \mathcal{L}^n$ of the lattice $\mathcal{L}$, however it is unclear whether the specialized version of their attack using vectors $s, (v_1, \dots, v_n)$ where $s - v_i \in \mathcal{L}$ is still applicable. It seems to be easier to recover the key when using multiple signatures from the same message as a lattice basis when using lattice reduction algorithms rather than using random small vectors of the lattice: this could imply that diagonal dominant basis have inner weaknesses beyond the simple instantiation of DRS. From our understanding, the secret key matrices they generated for their tests used a noise matrix with coefficients within $\{-1, 0, 1\}$, which could have had an impact in their experimentations. It is still unknown if other noise types such as the ones in DRS or the type of noise we are about to propose are affected: to the best of our knowledge, DRS was not quoted in their work.

We stress that we do not claim the new setup to be perfectly secure against Li, Liu, Nitaj and Pan’s attack, we merely claim more experimentations would need to be done as of now. Furthermore, the countermeasures proposed by Li, Liu, Nitaj and Pan also apply to those new keys, and should be applied if one wishes for a more concrete security. The next attack however does not have clear known countermeasures as of now and is the main focus of this paper.

3.3 Yu and Ducas’s attack on the DRS instantiation of the initial scheme of PKC’08

We explained in the previous section about the security of DRS against Li, Liu, Nitaj and Pan’s attack. On the other hand, it is unclear if such a modification would add an extra weakness against Yu and Ducas’s heuristic attack. Their attack work in two steps. The first one is based on recovering certain coefficients of a secret key vector using machine learning and statistical analysis. The second is classical lattice-reduction attack to recover the rest of the secret key.

For the first step, Yu and Ducas noticed that the coefficients B of the secret key and the 1 could be distinguished via machine learning techniques [30], noticing for one part that the non-diagonal coefficients follow an “absolute-circulant” structure, and the fact that only two types of non-zero values exist. Based on this information, a surprisingly small amount of selected “features” to specialize a “least-square fit” method allowed them to recover both positions and signs of all if not most coefficients B of a secret vector. We note they did not conduct an exhaustive search on all possible methods according to their paper thus stressing that their method might not be the best. We did not conduct much research on the related machine learning techniques therefore we cannot comment much on this part as of now.

On the second step, the recovered coefficients and their positions and signs allowed them to apply the Kannan embedding attack on a lattice with the exact same volume as the original public key but of a much lower dimension than the original authors of DRS based their security on, by scrapping the known B noise coefficients. Strictly speaking, using the same notation as in the previous description of DRS and assuming the diagonal coefficient is equal to the dimension, the initial search of a shortest vector of length $\sqrt{B^2 N_b + N_1 + 1}$ in a lattice of dimension n of determinant n^n becomes a search of a shortest vector of length $\sqrt{N_1 + 1}$ in a lattice of dimension $n - N_b$ of determinant n^n . The efficiency of lattice reduction techniques then affects the evaluation of the security strength of the original DRS scheme.

Yu and Ducas conducted experiments and validated their claims, reducing the security of the initial submission of DRS from 128-bits to maybe at most 80-bits, using BKZ-138. The original concept (not the instantiation) from [21], however, still seems to be safe for now: although it has no security proof, to the

best of our knowledge, no severe weaknesses have been found so far. Furthermore, Yu and Ducas advised of some potential countermeasures to fix DRS, i.e breaking the structure of the particular instance that was submitted: the deterministic approach of the number of B , 1, being limited to those 2 values (5 if we consider zeroes and signs), and the “absolute-circulant” structure. They also pointed that a lack of security proof could be problematic and gave some opinions about how one can potentially find provable security for the DRS scheme.

In the following section, we provide a countermeasure which follows some of the recommendations given by Yu and Ducas as breaking the secret key noise structure and giving some statistical heuristic, while still preserving the original idea given in PKC 2008 [21].

4 New setup

We do not change any algorithm here aside the setup of the secret key: the public key generation method is left unchanged, along with the signature and verification. Compared to the old scheme, this new version is now determined by less parameters, which leave 6 of them using the previous DRS: the dimension n , a random generator seed s , a signature bound D , a max norm for hashed messages δ , a sparsity parameter Δ that we always set to one, and R a security parameter determining the number of multiplication rounds to generate the public key.

We choose random noise among all the possible noises vectors which would still respect the diagonal dominant property of the secret key. This choice is following Yu and Ducas’s suggestions on breaking the set of secret coefficients, the “absolute-circulant” structure of the secret key, and allowing us to provide statistical evidence.

Although we want to have random noise, we must ensure we can still sign every message and thus guarantee the diagonal dominant structure of our secret key. Hence, the set of noise vectors we need to keep are all the vectors $v \in \mathbb{Z}^n$ that have a taxicab norm of $\|v\|_1 \leq D - 1$. Let us call that set V_n .

This new setup will also change the bounds used for the public key, as the original DRS authors linked several parameters together to ensure computations stay within 64-bits. However, our paper has a more theoretical approach and we do not focus on the technical implementations yet, which could be left for further work.

4.1 Picking the random vectors

We are aiming to build the new noise matrix M , which is a $n \times n$ matrix such that $M \in V_n^n$. In that regard, we construct a table we will call T with D entries such that

$$T[i] = \#\text{vectors } v \in V_n \text{ with } i \text{ zeroes.}$$

This table is relatively easy to build and does not take much time, one can for example use the formulas derivated from [24] and [10].

From this table, we construct another table T_S such that $T_S[k] = \sum_{i=0}^k T[i]$.

The generation algorithm of the table T_S , which we will use as a precomputation for our new setup algorithm can be seen in Alg 5.

```

Input: - all initial parameters;
Output: -  $T_S$  the table sum;
// Initialization
 $m \leftarrow \min(\text{dimension}, \text{diagonal value } D)$ ;
 $T \leftarrow \{1\}^{m+1}$ ;
 $T_S \leftarrow \{1\}^{m+1}$ ;
// Construct array T
// Construct array T : loop over the norm
for  $j = 2 ; j \leq D ; j = j + 1$  do
    // Construct array T : loop over the number of non-zeroes
    elements in each possibility
    for  $i = 2 ; i \leq m + 1 ; i = i + 1$  do
         $x \leftarrow 2^{i-1} \binom{n}{i-1} \binom{j-1}{i-2}$ ;
         $T[m + 1 - i] \leftarrow T[m + 1 - i] + x$ ;
    end
end
// Construct array  $T_S$  from T
for  $i = 1 ; i \leq m ; i = i + 1$  do
     $T[i + 1] \leftarrow T[i + 1] + T[i]$ ;
end
 $T_S \leftarrow T$ ;
return  $T_S$ ;

```

Algorithm 5: Secret key table precomputation

Let us denote the function $Z(x) \rightarrow y$ such that $T_S[y - 1] < x \leq T_S[y]$.

Since T_S is trivially sorted in increasing order $Z(x)$ is nothing more than a dichotomy search inside an ordered table.

If we pick randomly x from $[0; T_S[D - 1]]$ from a generator with uniform distribution $g() \rightarrow x$ then we got $Zero() \rightarrow Z(g(x))$ a function that selects uniformly an amount of zeroes amount all vectors of the set V_n , i.e.

$$Zero() \rightarrow \#\text{zeroes in a random } v \in V_n$$

Now that we can generate uniformly the number of zeroes we have to determine the coefficients of the non-zero values randomly, while making sure the final noise vector is still part of V_n . A method to give such a vector with chosen

taxicab norm is given in [27] as a correction of the Kraemer algorithm. As we do not want to choose the taxicab norm M directly but rather wants to have any random norm available, we add a slight modification: the method in [27] takes k non-zero elements $x_1, \dots, x_k$ such that $x_i \leq x_{i+1}$ and forces the last coefficient to be equal to the taxicab norm chosen, i.e $x_k = M$. By removing the restriction and using $x_k \leq D$, giving the amount of non-zero values, we modify the method to be able to take over any vector values in V_n with the help of a function we will call

$$\mathbf{KraemerBis}(z) \rightarrow \text{random } v \in V_n$$

such that v has z zeroes which is described in Alg 6

```

Input: - all initial parameters;
- a number of zeroes  $z$ ;
Output: - a vector  $v$  with  $z$  zeroes and a random norm inferior or equal to  $D$ ;
// Algorithm start
 $v \in \mathbb{N}^n$ ;
Pick randomly  $n - z + 1$  elements such that  $0 \leq x_0 < x_1 < \dots < x_{n-z} \leq D$ ;
for  $i = 1 ; i \leq n - z ; i = i + 1$  do
|  $v[i] \leftarrow x_i - x_{i-1}$ ;
end
for  $i = n - z + 1 ; i \leq n ; i = i + 1$  do
|  $v[i] \leftarrow 0$ ;
end
return  $v$ ;

```

Algorithm 6: KraemerBis

With both those new parts, the new setup algorithm we construct is presented in Alg 7 using Kraemer bis. We note that in our algorithm, the diagonal coefficient in the secret key is not guaranteed to be equal to the bound used for the maximum norm of the signatures. Nevertheless, we will show that the termination is still ensured in Sec 4.2. This heavy setup naturally affects the speed of the DRS setup, as we noticed in our experiments as shown in Sec 4.3.

4.2 A slightly more general termination proof

The proof stated in the DRS report on the NIST website [20] was considering that the diagonal coefficient of $S = d * I_d + M$ stayed equal to the signature bound, which is not our case. We show here that the reduction is still guaranteed nevertheless. Suppose that some coefficients of the noise matrix M are non-zero on the diagonal. Re-using for the most part notations of the original report, where:

- m is the message we want to reduce, which we update step by step

```

Input: - all initial parameters;
- another extra random seed  $x_2$ ;
Output: -  $x, S$  the secret key;
 $S \leftarrow D \times Id_n$ ;
 $t \in \mathbb{Z}^n$ ;
InitiateRdmSeed( $x_2$ );
for  $i = 1 ; i \leq n ; i = i + 1$  do
     $Z \leftarrow \text{Zero}()$ ;
     $t \leftarrow \text{KraemerBis}(Z)$ ;
    for  $j = 1 ; j \leq n - Z ; j = j + 1$  do
         $| t[j] \leftarrow t[j] \times \text{RdmSgn}()$ 
    end
     $t \leftarrow \text{RdmPmtn}(t)$ ;
     $S[i] \leftarrow S[i] + t$ ;
end
return  $x, S$ ;

```

Algorithm 7: New secret key generation (square matrix of dimension n)

- M is the noise matrix (so M_i is the i -th noise row vector).
 - d is the signature bound for which the condition $\|m\|_\infty < d$ has to be verified.
- We note d_i the i -th diagonal coefficient of the secret key S .

Obviously, the matrix will still be diagonal dominant in any case. Let us denote d_i the diagonal coefficient $S_{i,i}$ of $S = D - M$.

If $d > d_i$ we can use the previous reasoning and reduce $|m_i|$ to $|m_i| < d_i < d$, but keep in mind we stop the reduction at $|m_i| < d$ to ensure we do not leak information about the noise distribution.

Now $d_i > d$ for some i : reducing to $|m_i| < d_i$ is guaranteed but not sufficient anymore as we can reach $d < |m_i| < d_i \leq d + \Delta < 2d$. Let us remind that $\Delta = d - \sum_{j=1}^n |M_{i,j}|$, where Δ is strictly positive as an initial condition of the DRS signature scheme (both on the original submission and this paper), $d_i = d + c$ where $c = |M_{i,i}|$.

Without loss of generality as we can flip signs, let us set $m_i = d + k < d_i = d + c$ with $k \geq 0$ the coefficient to reduce. Subtracting by S_i transforms

$$m_i \leftarrow (d + k) - d_i = (d + k) - (d + c) = k - c < 0$$

with $d > c > k \geq 0$. Therefore the reduction of $\|m\|_1$ without the noise is

$$\|m\|_1 \leftarrow \|m\|_1 - (d + k) + (c - k) = \|m\|_1 - (d - c) - 2k.$$

but the noise contribution on other coefficients is at worst $(d - \Delta) - c$ thus

$$\begin{aligned} \|m\|_1 &\leftarrow \|m\|_1 - (d - c) - 2k + (d - c - \Delta). \\ \|m\|_1 &\leftarrow \|m\|_1 - 2k - \Delta = \|m\|_1 - (2k + \Delta). \end{aligned}$$

where $2k + \Delta > 0$. Therefore the reduction is also ensured in the case $d_i > d$.

4.3 Setup performance

Compared to the initial NIST submission where the code was seemingly made for clarity and not so much for performance, we wrote our own version of DRS using NIST specifications and managed to have much higher performance. However, most of the performance upgrade from the initial code have nothing much to do with the algorithms of the DRS scheme: we did notice that most of the time taken by the DRS initial code was used for the conversion from the character arrays to integer matrices and vice-versa, which they had to do to respect the NIST specifications: the algebraic computations themselves were actually reasonably fast, considering the size of the objects manipulated.

This is the reason why we decided to isolate the secret matrix generation code from the rest of the initial original DRS code, in order to have a fair comparison between our own secret key generation algorithm to theirs. In that regard we choose to compare similar matrix sizes instead of similar security, as initial security estimates for the DRS submission were severely undermined by Yu and Ducas’s recent discoveries and thus would lead to comparing efficiency on matrices with massively different sizes. Therefore we are making tests on the initial parameters of the DRS scheme. Looking purely at the secret key generation, we are indeed much slower, as shown in Table 1.

Table 1. Secret key generation time in seconds for 10^4 keys

<i>Dimension</i>	912	1160	1518
<i>OldDRS</i>	28.71	44.15	79.57
<i>NewDRS</i>	317.45	631.89	993.92

Note that we use the options $-march = native$ and $-Ofast$ which led us to use *AVX512* instructions and other *gcc* optimization tweaks. The new setup is barely parallelizable as there is almost no code that can be vectorized which also explains the huge difference.

5 Security estimates

5.1 BDD-based attack

The security is based on what is known as the currently most efficient way to attack the scheme, a **BDD**-based attack as described in Alg 8.

Currently, the most efficient way to perform this attack will be, first, to transform a **BDD** problem into a Unique Shortest Vector Problem (**uSVP**) (Kannan’s Embedding Technique [9]), assuming $v = (0, \dots, 0, d, 0, \dots, 0)$, and use


```

Input: public key  $Pk$  of full rank  $n$ , diagonal coefficient  $d$ ,  $\mathbf{BDD}_\gamma$  solver  $\phi$ 
Output: secret key  $Sk = (D - M)$ 
 $Sk \leftarrow d * Id_n$ ;
foreach  $\{i \in [1..n]\}$  do
  |  $r \leftarrow \phi(\mathcal{L}(Pk), Sk[i]), Sk[i] \leftarrow Sk[i] + r$ ;
end
return  $Sk$ ;

```

Algorithm 8: Diagonal Dominant Key recovery attack

lattice reduction techniques on the lattice spanned by $[v|1]$ and the rows of $[B|0]$. By using this method, we obtain a **uSVP** with a gap

$$\gamma \approx \frac{\Gamma\left(\frac{n+3}{2}\right)^{\frac{1}{n+1}} \text{Det}(\mathcal{L})^{\frac{1}{n+1}}}{\sqrt{\pi} \|M_1\|_2} \approx \frac{\Gamma\left(\frac{n+3}{2}\right)^{\frac{1}{n+1}} d^{n \frac{1}{n+1}}}{\sqrt{\pi} \|M_1\|_2}. \quad (1)$$

Lattice reduction methods are well studied and their strength are evaluated using the Hermite factor. Let $\mathcal{L}$ a d -dimensional lattice, the Hermite factor of a basis B of $\mathcal{L}$ is given by $\|B[1]\|_2 / \text{det}(\mathcal{L})^{\frac{1}{n}}$. Consequently, lattice reduction algorithms strengths are given by the Hermite factor of their expected output basis. In [6], it was estimated that lattice reduction methods solve uSVP_γ with γ a fraction of the Hermite factor. We will use a conservative bound of $\frac{1}{4}$ for the ratio of the uSVP gap to the Hermite factor. As we do not have a fixed euclidean norm for our secret vectors we have to rely on the approximates given to us by our new random method in sampling noise vectors M_i . In our case, we know that for any vector $v \in \mathbb{Z}^n$ we have $\|v\|_2 \geq \frac{\|v\|_1}{\sqrt{n}}$, and our experiments (as seen below) allow us to use a higher bound $\|v\|_2 \gtrsim \sqrt{2} \frac{\|v\|_1}{\sqrt{n}}$.

5.2 Expected Security Strength

Different papers are giving some relations between the Hermite factor and the security parameter λ [22,8] often using BKZ simulation [4]. Aiming to be conservative, we are to assume a security of $2^{128}, 2^{192}, 2^{256}$ for a Hermite factor of $1.006^d, 1.005^d, 1.004^d$, respectively. We set $D = n$, pick hashed messages $h(m)$ such that $\log_2(\|h(m)\|_\infty) = 28$, $R = 24$ and $\Delta = 1$.

Table 2. Parameter Sets.

Dimension	Δ	R	δ	γ	2^λ
1108	1	24	28	$< \frac{1}{4}(1.006)^{d+1}$	2^{128}
1372	1	24	28	$< \frac{1}{4}(1.005)^{d+1}$	2^{192}
1779	1	24	28	$< \frac{1}{4}(1.004)^{d+1}$	2^{256}

Table 2 parameters have been chosen to obtain a uSVP gap (Equation 1) with $\gamma < \frac{\delta^{d+1}}{4}$ for $\delta = 1.006, 1.005, 1.004$. Our experiments show us that the distribution of zeroes among sampled noise vectors form a Gaussian and so does the euclidean norm of noise vectors when picking our random elements x, x_i uniformly. Here we include below the distribution of 10^6 randomly generated noise vectors v with the x-axis representing $f(v) = \lfloor 100\sqrt{\frac{\|v\|_2^2}{D}} \rfloor$ where D is the signature bound (see Fig 1).

Fig. 1. $f(v)$ distribution for $n = 1108, 1372, 1779$ and $D = n - 1$ over 10^6 samples

We can see that the generated noise vectors follow a Gaussian distribution as far as their norms are concerned, and we believe it makes guessing values much harder for an attacker should they choose to focus on finding specific values or vectors (as it was the case in the original attack from Yu and Ducas [30]). We also conducted experiments, using BKZ20 from the fplll library [28] (see Fig 2). Without any surprise we notice our new setup is seemingly resistant around dimension 400, where conservative bounds led us to believe the break happen until approximately dimension 445. However the sample size is relatively small (yet computationally expensive to obtain) and thus should not be taken as a proof value, but rather as a heuristic support against heuristic attacks.

6 Conclusion and open questions

We presented in this paper a new method to generate secret keys for the DRS scheme, providing experimental results on the statistical distribution of the keys generated. We demonstrate that our new approach is sufficient to improve DRS to be secure against machine learning attacks as reported earlier in the literature. However, the secret matrix is still diagonal dominant and it remains an open question whether there exists a tight security proof to a well-known problem or if there is any unforeseen weaknesses to diagonal dominant lattices as both Li, Liu, Nitaj and Pan's [11] and Yu and Ducas's attacks [30] could lead to. The open questions for improvement stated in the original DRS report are also still applicable to our proposed iteration.

On the technical side, our method to generate random samples is also slow and might need improvement. It also impacts the setup as mentioned earlier, as

Fig. 2. Percentage of key recoveries of BKZ20 (20 sample keys/dim)

keeping the current DRS parameters one can see the possibility to overflow and go over 64-bits, even though the probability is extremely low, thus changing the public key generation is also left as an open question. The initial DRS scheme was very conservative not only on their security but also the manipulated integer size bounds: one might use heuristics to drastically increase the memory efficiency of the scheme.

Acknowledgements We would like to thank Yang Yu, Léo Ducas and the anonymous reviewers for useful comments and suggestions.

References

1. Ajtai, M.: Generating hard instances of lattice problems. In: STOC '96. pp. 99–108. ACM (1996)
2. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: STOC '97. pp. 284–293. ACM (1997)
3. Brualdi, R.A., Ryser, H.J.: Combinatorial matrix theory, vol. 39. Cambridge University Press (1991)
4. Chen, Y., Nguyen, P.Q.: Bkz 2.0: Better lattice security estimates. In: ASIACRYPT 2011. pp. 1–20. Springer (2011)
5. Fischlin, R., Seifert, J.P.: Tensor-based trapdoors for cvp and their application to public key cryptography. In: Cryptography and Coding, pp. 244–257. Springer (1999)
6. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: EUROCRYPT 2008, pp. 31–51. Springer (2008)
7. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: CRYPTO'97, pp. 112–131. Springer (1997)

8. Hoffstein, J., Pipher, J., Schanck, J.M., Silverman, J.H., Whyte, W., Zhang, Z.: Choosing parameters for ntruencrypt. In: CT-RSA 2017. pp. 3–18. Springer (2017)
9. Kannan, R.: Minkowski’s convex body theorem and integer programming. *Mathematics of operations research* **12**(3), 415–440 (1987)
10. Knuth, D.E., Graham, R.L., Patashnik, O., et al.: *Concrete mathematics*. Addison Wesley (1989)
11. Li, H., Liu, R., Nitaj, A., Pan, Y.: Cryptanalysis of the randomized version of a lattice-based signature scheme from pkc’08. In: ACISP 2018. pp. 455–466. Springer (2018)
12. Liu, M., Wang, X., Xu, G., Zheng, X.: Shortest lattice vectors in the presence of gaps. *IACR Cryptology ePrint Archive* **2011**, 139 (2011)
13. Lyubashevsky, V., Micciancio, D.: On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: CRYPTO 2009, pp. 577–594. Springer (2009)
14. Micciancio, D.: Improving lattice based cryptosystems using the hermite normal form. In: *Cryptography and Lattices*, pp. 126–145. Springer (2001)
15. Minkowski, H.: *Geometrie der Zahlen*. B.G. Teubner, Leipzig (1896)
16. Nguyen, P.: Cryptanalysis of the goldreich-goldwasser-halevi cryptosystem from crypto’97. In: CRYPTO’99. pp. 288–304. Springer (1999)
17. NIST: Post-quantum cryptography standardization (2018), <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography>
18. NIST: Nist kicks off effort to defend encrypted data from quantum computer threat (28/04/2016), www.nist.gov/news-events/news-updates/
19. Paeng, S.H., Jung, B.E., Ha, K.C.: A lattice based public key cryptosystem using polynomial representations. In: PKC 2003, pp. 292–308. Springer (2003)
20. Plantard, T., Sipasseuth, A., Dumondelle, C., Susilo, W.: Drs : Diagonal dominant reduction for lattice-based signature. PQC Standardization Conference, Round 1 submissions (2018), <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/DRS.zip>
21. Plantard, T., Susilo, W., Win, K.T.: A digital signature scheme based on cvp max. In: PKC 2008. pp. 288–307. Springer (2008)
22. van de Pol, J., Smart, N.P.: Estimating key sizes for high dimensional lattice-based systems. In: IMACC 2013. pp. 290–303. Springer (2013)
23. Regev, O.: New lattice-based cryptographic constructions. *Journal of the ACM (JACM)* **51**(6), 899–942 (2004)
24. Serra-Sagristà, J.: Enumeration of lattice points in l_1 norm. *Information processing letters* **76**(1-2), 39–44 (2000)
25. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* **26**(5) (1997)
26. Sloane, N.J.A.: *Encrypting by Random Rotations*, pp. 71–128. Springer Berlin Heidelberg, Berlin, Heidelberg (1983)
27. Smith, N.A., Tromble, R.W.: *Sampling uniformly from the unit simplex* (2004)
28. The FPLLL team: Fplll, a lattice reduction library, <https://github.com/fplll>
29. Wei, W., Liu, M., Wang, X.: Finding shortest lattice vectors in the presence of gaps. In: CT-RSA 2015. pp. 239–257. Springer (2015)
30. Yu, Y., Ducas, L.: Learning strikes again: The case of the drs signature scheme. In: ASIACRYPT 2018. pp. 525–543. Springer (2018)