Learning Strikes Again: the Case of the DRS Signature Scheme *

Léo Ducas[1] and Yang Yu[2,3]

[1] Cryptology Group, CWI, Amsterdam, The Netherlands
[2] University of Rennes 1, Rennes, France
[3] IRISA, Rennes, France

Abstract. Lattice signature schemes generally require particular care when it comes to preventing secret information from leaking through signature transcript. For example, the Goldreich-Goldwasser-Halevi (GGH) signature scheme and the NTRUSign scheme were completely broken by the parallelepiped-learning attack of Nguyen and Regev (Eurocrypt 2006). Several heuristic countermeasures were also shown vulnerable to similar statistical attacks.

At PKC 2008, Plantard, Susilo and Win proposed a new variant of GGH, informally arguing resistance to such attacks. Based on this variant, Plantard, Sipasseuth, Dumondelle and Susilo proposed a concrete signature scheme, called DRS, that is in the round 1 of the NIST post-quantum cryptography project.

In this work, we propose yet another statistical attack and demonstrate a weakness of the DRS scheme: one can recover some partial information of the secret key from sufficiently many signatures. One difficulty is that, due to the DRS reduction algorithm, the relation between the statistical leak and the secret seems more intricate. We work around this difficulty by training a statistical model, using a few features that we designed according to a simple heuristic analysis.

While we only recover partial secret coefficients, this information is easily exploited by lattice attacks, significantly decreasing their complexity. Concretely, we claim that, provided that 100 000 signatures are available, the secret key may be recovered using BKZ-138 for the first set of DRS parameters submitted to the NIST. This puts the security level of this parameter set below 80-bits (maybe even 70-bits), to be compared to an original claim of 128-bits.

Furthermore, we review the DRS v2 scheme that is proposed to resist above statistical attack. For this countermeasure, while one may not recover partial secret coefficients exactly by learning, it seems feasible to gain some information on the secret key. Exploiting this information, we can still effectively reduce the cost of lattice attacks.

Keywords: Cryptanalysis · Lattice based signature · Statistical attack · Learning · BDD

* This is an extended version of the conference paper [36]. New material has been added as Section 6, treating the application of our technique to the second version of DRS [30, 33].

1 Introduction

At Crypto'97, Goldreich, Goldwasser and Halevi proposed the encryption and signature schemes [18] whose security relies on the hardness of lattice problems. Concurrently, a practical scheme, NTRUEncrypt was proposed, and adapted for signatures a few years later (NTRUSign [21]). In 2006, Nguyen and Regev presented a celebrated statistical attack [26] and completely broke GGH and NTRUSign in practice. The starting point of NR attack is a basic observation that any difference between signature and message always lies in the parallelepiped spanned by secret key. Thus each signature leaks partial information about the secret key, which allows to fully recover the secret key from sufficiently many signatures. In 2012, Ducas and Nguyen revisited NR attack [14] and showed that it could be generalized to defeat several heuristic countermeasures [21, 22].

Designing secure and efficient lattice based signatures remains a challenging problem. To get rid of information leaks, the now standard method is to use a delicate sampling algorithm for trapdoor inversion [17, 28].[4] Following such setting, it can be proved that signatures are independent of the secret key. Yet this provable guarantee doesn't come cheap in terms of efficiency and simplicity: it remains very tempting to make more aggressive design choices.

Such a design was proposed by Plantard, Susilo and Win [31]. It is very close to the original GGH scheme, with a modified reduction algorithm that produces signatures falling in a known hypercube, independent of the secret key. According to the authors, such property should prevent the NR attack. The main idea of [31] is to reduce vectors under ℓ_∞-norm instead of Euclidean norm. Plantard, Sipasseuth, Dumondelle and Susilo then updated this scheme, and submitted it to the NIST post-quantum cryptography project, under the name of DRS [29], standing for Diagonal-dominant Reduction Signature. DRS is in the list of round 1 submissions to the NIST post-quantum cryptography project.

Our results. In this work, we present a statistical attack against the DRS scheme [31, 29]. We first notice that while the support of the transcript distribution is indeed fixed and known, the distribution itself is not, and is related to the secret key. More concretely, in the DRS signature, the reduction algorithm will introduce some correlations among coordinates w_i's of the signature, and these correlations are strongly related to certain coefficients of the secret key $\mathbf{S}$.

In more details, we assume that the coefficient $\mathbf{S}_{i,j}$ can be well approximated by some function of the distribution of (w_i, w_j) and proceed to profile such a function according to known instances (the training phase). Once we have the function, we can measure over sufficient signatures and obtain the guess for an unknown secret $\mathbf{S}$.

With a few extra amplification tricks, we show this attack to be rather effective: for the first set of parameters, 100 000 signatures suffice to locate all the

[4] Alternatively, one may resort to the (trapdoorless) Fiat-Shamir with aborts approach such as done in [24, 13], yet for simplicity, we focus our discussion on the Hash-then-Sign approach.

large coefficients of the secret matrix $\mathbf{S}$ and to determine most of their signs as well. Finally, we can feed this leaked information back into lattice attacks (BDD-uSVP attack), significantly decreasing their cost. Concretely, we claim that the first set of parameters offers at most 80-bits of security, significantly less than the original claim of 128-bits.

As a by-product, we formalize how to accelerate BDD attack when given some known coefficients of the solution. More specifically, we are able to construct a lattice of the same volume but smaller dimension for this kind of BDD instances.

To resist the above attack, Sipasseuth, Plantard and Susilo proposed the DRS v2 scheme [30, 33] as a countermeasure. In this paper, we also analyze DRS v2. Indeed this scheme has a better resistance to the aforementioned attack: we cannot determine partial secret coefficients by learning directly. Nevertheless it seems still feasible to make a guess carrying secret information using learning technique. Provided sufficiently many samples are available, the guess can be effectively exploited by lattice attacks. As a consequence, we claim that the parameter set for at least 128-bits of security actually provides the security of at most 100-bits once 2^{30} transcripts are released.

Our scripts are open source for checking, reproduction or extension purposes, available at https://github.com/yuyang-Tsinghua/DRS_Cryptanalysis.

Related work. In 2018, Li, Liu, Nitaj and Pan proposed a chosen message attack [20] against the randomized version of Plantard-Susilo-Win GGH signature variant [31]. Their starting observation is that the difference between two signatures of a same message is a relatively short lattice vector in the randomized Plantard-Susilo-Win scheme, then from enough such short lattice vectors one may recover some short vectors of the secret matrix by lattice reduction. The randomized modification is a crucial weakness of Plantard-Susilo-Win scheme exploited by the attack in [20]. To fix such weakness, the authors mentioned two strategies: storing previous messages and padding a random nonce in the hash function. In comparison, our targeted scheme and technical idea are different from those in [20]. More importantly, the weakness of the DRS scheme that we demonstrate does not seem to be easily fixed.

Roadmap. In Section 2, we introduce notations and background on lattices. In Section 3, we provide a brief description of DRS signature scheme. Then we explain how to learn large coefficients of the secret matrix in Section 4, and how to combine partial information and lattice techniques to recover the full key in Section 5. In Section 6, we provide a cryptanalysis of the DRS v2 scheme. Finally, we conclude and discuss potential countermeasure in Section 7.

Acknowledgements. We thank Thomas Plantard, Arnaud Sipasseuth, Han Zhao and Paul Kirchner for helpful discussions and comments. We are also grateful to Yanbin Pan for sharing their work. Léo Ducas is supported by a Veni Innovational Research Grant from NWO under project number 639.021.645 and by the European Union H2020 Research and Innovation Program Grant 780701 (PROMETHEUS). This work has also received a French government support granted to the CominLabs excellence laboratory and managed by the National Research Agency in the “Investing for the Future” program under the project TYREX, reference ANR-10-LABX-07-01.

2 Preliminaries

We use bold lowercase letters for vectors and denote by v_i the i-th entry of the vector $\mathbf{v}$. We denote by $\|\mathbf{v}\|$ (resp. $\|\mathbf{v}\|_1$ and $\|\mathbf{v}\|_\infty$) the Euclidean norm (resp. ℓ_1-norm and ℓ_∞-norm) of $\mathbf{v}$. For consistency with the implementation, we start indexing vectors from 0: the subscript of each entry of $\mathbf{v} \in \mathbb{R}^n$ is to be interpreted as an element of $\mathbb{Z}_n = \{0, \cdots, n-1\}$.

Let $\mathbf{rot}_i(\mathbf{v}) = (v_{-i}, \cdots, v_{-i+n-1})$ be a rotation of $\mathbf{v} \in \mathbb{R}^n$. We denote by $\mathbf{srot}_i(\mathbf{v})$ (for signed-rotation) the random variable generated by flipping the signs of all entries of $\mathbf{rot}_i(\mathbf{v})$ independently at random with probability 1/2. We define the set

$$\mathcal{T}(n, b, N_b, N_1) = \left\{ \mathbf{v} \in \mathbb{Z}^n \,\middle|\, \mathbf{v} \text{ has exactly } \begin{array}{l} N_b \text{ entries equal to } b; \\ N_1 \text{ entries equal to } 1; \\ n - N_1 - N_b \text{ entries equal to } 0. \end{array} \right\}.$$

We use bold capital letters for matrices and denote by $\mathbf{v}_i$ the i-th row of the matrix $\mathbf{V}$, i.e. $\mathbf{V} = (\mathbf{v}_0, \cdots, \mathbf{v}_{n-1})$. We use $\mathbf{V}_{i,j}$ to represent the entry in the i-th row and j-th column of $\mathbf{V}$. Let $\mathbf{I}_n$ be the n-dimensional identity matrix. We denote by $\mathbf{ROT}(\mathbf{v})$ (resp. $\mathbf{SROT}(\mathbf{v})$) the matrix $(\mathbf{rot}_0(\mathbf{v}), \cdots, \mathbf{rot}_{n-1}(\mathbf{v}))$ (resp. $(\mathbf{srot}_0(\mathbf{v}), \cdots, \mathbf{srot}_{n-1}(\mathbf{v}))$). Note that all $\mathbf{srot}_i(\mathbf{v})$'s in $\mathbf{SROT}(\mathbf{v})$ are generated independently. A matrix $\mathbf{V}$ is diagonal dominant if $\mathbf{V}_{i,i} > \sum_{j \neq i} |\mathbf{V}_{i,j}|$ for all i.

For a distribution D, we write $X \leftarrow D$ when the random variable X is sampled from D. Given a finite set S, let $U(S)$ be the uniform distribution over S. We denote by $\mathbb{E}(X)$ the expectation of random variable X.

A (full-rank) n-dimensional lattice $\mathcal{L}$ is the set of all integer combinations of linearly independent vectors $\mathbf{b}_1, \cdots, \mathbf{b}_n \in \mathbb{R}^n$, i.e. $\mathcal{L} = \{\sum_{i=1}^n x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\}$. We call $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$ a basis of $\mathcal{L}$ and write $\mathcal{L} = \mathcal{L}(\mathbf{B})$. For a unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$, we have $\mathbf{UB}$ is also a basis of $\mathcal{L}(\mathbf{B})$, i.e. $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{UB})$. Let $(\mathbf{b}_0^*, \cdots, \mathbf{b}_{n-1}^*)$ be the Gram-Schmidt vectors of $\mathbf{B}$. The volume of the lattice $\mathcal{L}(\mathbf{B})$ is $\mathrm{vol}(\mathcal{L}(\mathbf{B})) = \prod_i \|\mathbf{b}_i^*\|$ that is an invariant of the lattice. Given $\mathcal{L} \subseteq \mathbb{R}^n$ and $\mathbf{t} \in \mathbb{R}^n$, the distance between $\mathbf{t}$ and $\mathcal{L}$ is $\mathrm{dist}(\mathbf{t}, \mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L}} \|\mathbf{t} - \mathbf{v}\|$.

Lattice reduction is an important tool for solving lattice problems and estimating the security of lattice-based cryptosystems. The goal of lattice reduction is to find a basis of high quality. The quality of a basis $\mathbf{B}$ is related to its root Hermite factor $\mathbf{rhf}(\mathbf{B}) = \left(\frac{\|\mathbf{b}_0\|}{\mathrm{vol}(\mathcal{L}(\mathbf{B}))^{1/n}} \right)^{1/n}$. Currently, the most practical lattice reduction algorithms are BKZ [32] and BKZ 2.0 [11]. We denote by BKZ-β the BKZ/BKZ 2.0 with blocksize β. In general, we assume the root Hermite factor of a BKZ-β basis is bounded by

$$\delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e} \right)^{\frac{1}{2(\beta-1)}}$$

when $n \gg \beta > 50$.

3 The DRS Signature Scheme

In this section, we provide a brief description of the first DRS scheme. We may omit some details that are unnecessary for understanding our attack. For more details on the algorithms and implementations we refer to [29].

To start with, we introduce several public parameters of DRS:

- n : the dimension
- D : the diagonal coefficient of the secret key
- b : the magnitude of the large coefficients (i.e. $\{\pm b\}$) in the secret key
- N_b : the number of large coefficients per vector in the secret key
- N_1 : the number of small coefficients (i.e. $\{\pm 1\}$) per vector in the secret key

Following the setting provided in [29], the parameter D is chosen to be n and satisfies that $D > b \cdot N_b + N_1$.

The secret key of DRS is a matrix

$$\mathbf{S} = D \cdot \mathbf{I}_n - \mathbf{M}$$

where $\mathbf{M} = \mathbf{SROT}(\mathbf{v})$ with $\mathbf{v} \leftarrow U\left(\mathcal{T}(n, b, N_b, N_1) \bigcap \{\mathbf{v} \in \mathbb{Z}^n \mid v_0 = 0\}\right)$ is the noise matrix. It is easily verified that $\mathbf{S}$ is diagonal dominant. The public key is a matrix $\mathbf{P}$ such that $\mathcal{L}(\mathbf{P}) = \mathcal{L}(\mathbf{S})$ and the vectors in $\mathbf{P}$ are much longer than those in $\mathbf{S}$.

Hash space. The specification submitted to the NIST [29] is rather unclear about the message space. Namely, only a bound of 2^{28} is mentioned, which suggests a hash space $\mathcal{M} = (-2^{28}, 2^{28})^n$, following the original scheme [31]. Yet, we noted that the implementation seems to instead use the message space $\mathcal{M} = (0, 2^{28})^n$: the sign randomization is present, but commented out. Discussion with the designers[5] led us to consider this as an implementation bug, and we therefore focus on the analysis with $\mathcal{M} = (-2^{28}, 2^{28})^n$, following both the original scheme [31] and the intention of [29].

We strongly suspect that taking $\mathcal{M} = (0, 2^{28})^n$ would not be an effective countermeasure against the kind attack analyzed in this paper. Preliminaries experiments on this variant suggested that leak was stronger, but its relation to the secret key seemed more intricate.

For our experiments, we generated directly uniform points in that space rather than hashing messages to this space; according to the Random Oracle Model, this should make no difference.

[5] https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/DRS-official-comment.pdf

Signature. The signature algorithm of DRS follows the one in [31] and its main component is a message reduction procedure in ℓ_∞-norm. It is summarized below as Algorithm 1.

Algorithm 1: Message reduction in DRS signature algorithm

```
Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$
Output: a reduced message $\mathbf{w} \in \mathbb{Z}^n$ such that $\mathbf{w} - \mathbf{m} \in \mathcal{L}(\mathbf{S})$
$\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0, k \leftarrow 0$
repeat
    $q \leftarrow \lfloor w_i / D \rfloor_{\to 0}$,                (Rounding toward 0)
    if $q \neq 0$ then
        $\mathbf{w} \leftarrow \mathbf{w} - q\mathbf{s}_i$
        $k = 0$
    end if
    $k \leftarrow k + 1$, $i \leftarrow (i + 1) \bmod n$
until $k = n$
return $\mathbf{w}$
```

In brief, the message reduction is reducing successively each large coefficient m_i of the message $\mathbf{m}$ by qD such that $|m_i - qD| < D$ but adding $\pm q, \pm qb$ to m_j with $j \neq i$ according to the entries of $\mathbf{M}$, until all coefficients of the reduced message are within $(-D, D)$. Since $\mathbf{S}$ is diagonal dominant, the message can be reduced within bounded steps as proved in [29, 31].

Besides the reduced message $\mathbf{w}$, an auxiliary vector $\mathbf{k}$ is also included in the signature and used to accelerate the verification. To verify the signature, one would first check whether $\|\mathbf{w}\|_\infty < D$ and then check whether $\mathbf{m} - \mathbf{w} = \mathbf{kP}$. In later discussions, we shall ignore the auxiliary vector, because it can be calculated in polynomial time from $\mathbf{w}, \mathbf{m}$ and the public key $\mathbf{P}$.

4 Learning Coefficients of the Secret Matrix

All DRS signatures $\mathbf{w}$ lie in the region $(-D, D)^n$. Unlike the GGH scheme, the signature region is a known hypercube and independent of the secret matrix, thus the DRS scheme was deemed to resist statistical attacks. However, the distribution of random signature in $(-D, D)^n$ may be still related to the secret key, which would leak some key information.

In later discussion, we aim at a concrete parameter set

$$(n, D, b, N_b, N_1) = (912, 912, 28, 16, 432)$$

that is submitted to the NIST and claimed to provide at least 128-bits of security in [29].

4.1 Intuition on a potential leak

Our approach is to try to recover $\mathbf{S}_{i,j}$ by studying the distribution $W_{i,j}$ of (w_i, w_j). Indeed, when a reduction happens at index i: $\mathbf{w} \leftarrow \mathbf{w} - q\boldsymbol{s}_i$, and when $\mathbf{S}_{i,j} \neq 0$ some correlation is introduced between w_i and w_j. Symmetrically, correlation is also introduced when $\mathbf{S}_{j,i} \neq 0$. Another source of correlations is created by other reductions at index $k \notin \{i, j\}$ when both $\mathbf{S}_{k,i}$ and $\mathbf{S}_{k,j}$ are non-zero; these events create much less correlations since the diagonal coefficients are much larger, but those correlations accumulate over many k's. One is tempted to model the accumulated correlations as those of some bi-variate Gaussians with a certain covariance.

Of course, there are complicated "cascading" phenomena: by modifying a coefficient, a reduction may trigger another reduction at an other index. But let us ignore such phenomena, and just assume that several reductions at indices $k \neq i, j$ occur, followed by one reduction at index i with $q = \pm 1$, before the algorithm terminates. We depict our intuition as Figure 1.

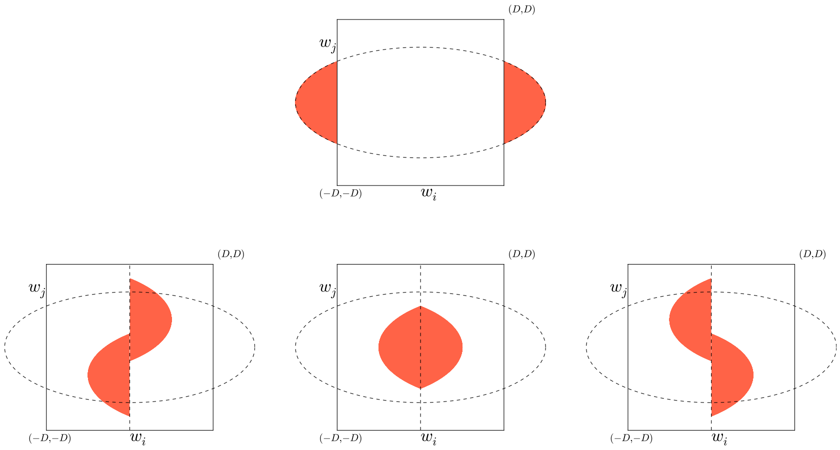

Fig. 1. Figures in the second row show the regions to which (w_i, w_j) in two cap regions will be moved by reduction at index i when $\mathbf{S}_{i,j} = -b, 0, b$ respectively from left to right.

In this simple model, we note that there are 4 degrees of liberty, 3 for the shape of the ellipsoid, and 1 for $\mathbf{S}_{i,j} = -b, 0, b$.[6] Therefore, one may expect to be able to recover all the parameters using 4 statistical measures. One natural choice is the following. First, measure the covariance matrix of the whole distribution, which gives 3 parameters. Assuming the clipped caps have small weights, this

[6] In fact, two of those degrees are fixed by the shape of the secret matrix: each rows of $\mathbf{S}$ has fixed Euclidean length, fixing the variance of w_i and w_j.

would roughly give the shape of the ellipsoid. For the last measure, one would select only sample for which $|w_i|$ is small, so as to focus on the superimposed displaced caps. With a bit of effort one would find an appropriate measurement.

Unfortunately, it seems rather hard to determine mathematically what will precisely happen in the full reduction algorithm, and to construct by hand a measurement on the distribution of (w_i, w_j) directly giving $\mathbf{S}_{i,j}$, i.e. a function f such that $f(W_{i,j}) = \mathbf{S}_{i,j}$.

4.2 Training

While constructing such a function f by a mathematical analysis may be hard, our hope is that such function may be easy to learn using standard techniques, ranging from least-square method to convolutional neural networks. Indeed, going back to Figure 1, recovering $\mathbf{S}_{i,j}$ from $W_{i,j}$ can essentially be viewed as a grey-scale image classification problem (the lightness of the pixel (x, y) corresponding to the density of $W_{i,j}$ at (x, y)).

Features. We therefore proceed to design a few *features*, according to the intuition built above. The average of each w_i is supposed to be 0, thus we do not treat it as a feature. Certainly, the covariance information is helpful, but we also introduce extra similar statistics to allow the learning algorithm to handle extra perturbations not captured by our simple intuition. We restrict our features to being symmetric: a sample (x, y) should have the same impact as $(-x, -y)$. Indeed, while quite involved, the whole reduction process preserves this symmetry.

More specifically, by scaling a factor of D, consider the distribution W to have support $(-1, 1)^2$. For a function f over $(-1, 1)^2$, we write $f(W) = \mathbb{E}_{(x,y)\leftarrow W}(f(x))$. The features mentioned before are listed below[7]:

- $f_1(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(x \cdot y)$;
- $f_2(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(x \cdot |x|^{1/2} \cdot y)$;
- $f_3(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(x \cdot |x| \cdot y)$.

We could go on with higher degrees, but this would cause some trouble. First, higher degree moments converge much slower. Secondly, taking too many features would lead to over-fitting.

Then, following our intuition, we want to also consider features that focus on the central region. Still, we do not want to give too much weight to samples with x very close to 0. Indeed, there will be some extra perturbation after the reduction at index i, which could flip the sign of x. A natural function to take this into account is the following.

- $f_4(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(x(x-1)(x+1) \cdot y)$. [8]

[7] We introduced a re-normalization factor D in our experiments. We keep it in this paper for consistency.

[8] As we are only going to consider linear models in our features, we could equivalently replace this feature by $\mathbb{E}_{(x,y)\leftarrow W}(x^3 \cdot y)$ because of the presence of f_1.

The most contributing sample will be the one for which $x = \pm 1/\sqrt{3}$, and it is not clear that this is the optimal range to select. We therefore offer to the learning algorithm a few variants of the above that select samples with smaller values of x, hoping that it can find a good selection by combining all of them:

- $f_5(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(2x(2x-1)(2x+1) \cdot y \mid |2x| \leq 1)$;
- $f_6(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(4x(4x-1)(4x+1) \cdot y \mid |4x| \leq 1)$;
- $f_7(W) = D \cdot \mathbb{E}_{(x,y)\leftarrow W}(8x(8x-1)(8x+1) \cdot y \mid |8x| \leq 1)$.

For any function f over $\mathbb{R}^2$, we call $f^t : (x, y) \mapsto f(y, x)$ the transpose of f. So far, we have introduced 13 different features, i.e. $f_1, \cdots, f_7$ and their transposes $f_8 = f_2^t, \cdots, f_{13} = f_7^t$.[9] We plot these functions in Figure 2.

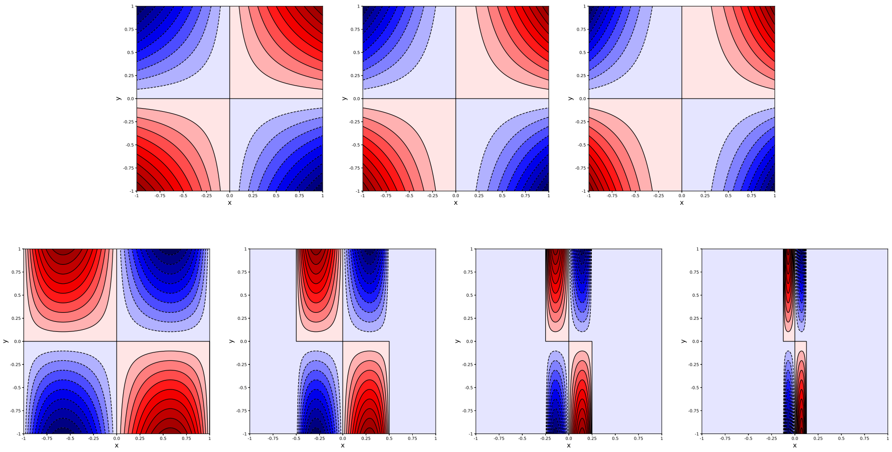

Fig. 2. The color matrices for $f_1, \cdots, f_7$. For any pixel at (x, y), its color is red (full-line contour), blue (dashed-line contour) when $f_i(x, y) > 0, \leq 0$ respectively. The deeper the color is, the larger $|f_i(x, y)|$ is.

Generating data. Then, we proceed to measure each $W_{i,j}$ for known values of $\mathbf{S}_{i,j}$, say, using 400 000 samples for each key $\mathbf{S}$, and using 30 different keys $\mathbf{S}$. This is implemented by our script `gen_training.py`. This took about 38 core-hours.

Training. We naturally considered using advanced machine learning techniques (support vector regression [8], random forest regression [23] and artificial neural networks) to construct a model, with the precious support of Han Zhao. Despite some effort, he was unable to find a method that outperforms what we achieved with a linear models $f = \sum_{\ell=1}^{13} x_\ell f_\ell$ trained using the *least-square fit* method. Yet his exploration was certainly far from exhaustive, and we do not conclude that least-square fit is the best method.

[9] Since f_1 is a symmetric function of (w_i, w_j), we did not count its transpose.

Evaluating and refining our model. After preliminary experiments, we noted that, depending on their position $(i-j)$, some coefficients $\mathbf{S}_{i,j}$ seem easier to learn than others. In this light, it is not clear that one should use the same function f for all indices i, j. Instead, we constructed two functions $f^+ = \sum x_\ell^+ f_\ell$, $f^- = \sum x_\ell^- f_\ell$ respectively for indices such that $i - j \bmod n \geq n/2$ and $i - j \bmod n < n/2$. The model obtained by the least-square fit method is provided in Table 1 and plotted in Figure 3. Moreover, the distributions of $f^+(W_{i,j})$, $f^-(W_{i,j})$ for $\mathbf{S}_{i,j} = \pm b, \pm 1, 0$ are illustrated in Figures 4 and 5.

Remark 1. For other set of parameters, or even to refine our attack and recover more secret information, it is of course possible to cut our modeling in more than 2 pieces, but this requires more training data, and therefore more computational resources.

i	1	2	3	4	5	6	7
x_i^-	-48.3640	354.9788	-289.1598	58.7149	-3.7709	-2.9138	2.3777
i		8	9	10	11	12	13
x_i^-		-21.2574	6.6581	3.5598	1.0255	0.4835	-0.3637
i	1	2	3	4	5	6	7
x_i^+	-67.9781	324.8442	-248.7882	44.6268	-4.1116	-2.6163	2.8288
i		8	9	10	11	12	13
x_i^+		-9.0923	3.1639	-0.8145	0.5204	0.3486	0.4920

Table 1. The model trained from 30 keys and 400 000 signatures per key. This is implemented by our script `gen_model.py`.

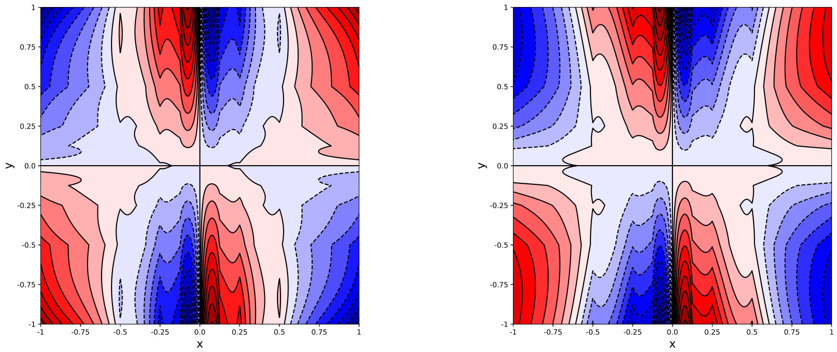

Fig. 3. The left graph is the color matrix for f^-, and the right one is for f^+.

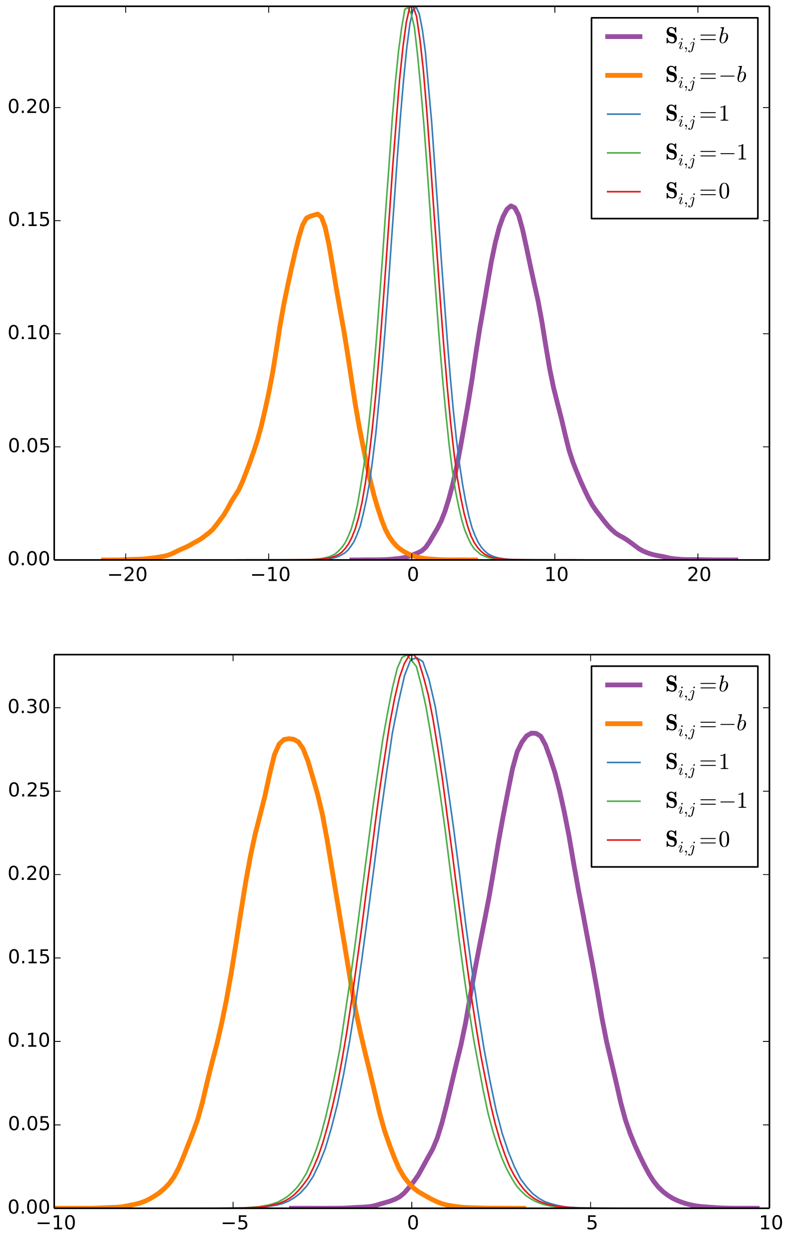

Fig. 4. The distributions of $f^{-}(W_{i,j})$, $f^{+}(W_{i,j})$ for $\mathbf{S}_{i,j} = \pm b, \pm 1, 0$. The upper one corresponds to f^{-} and the lower one corresponds to f^{+}. Experimental values measure over 20 instances and 400 000 samples per instance.

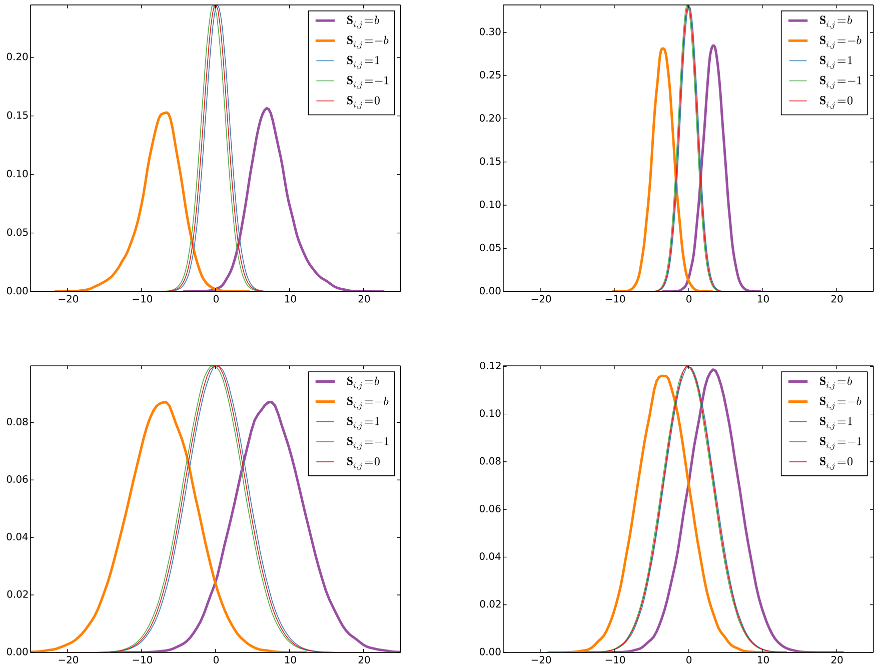

Fig. 5. The impact from sample sizes on the measured distributions of $f^{-}(W_{i,j})$, $f^{+}(W_{i,j})$. The left graphs correspond to f^{-} and the right graphs correspond to f^{+}. The upper graphs measure over 20 instances and 400 000 samples per instance, and the lower graphs measure over 20 instances and 50 000 samples per instance.

Remark 2. As shown in Figures 4 and 5, predicted values $f(W_{i,j})$ for large coefficients are usually of larger size than those for $-1, 0, 1$. Compared with large coefficients far from the main diagonal, those near the main diagonal tend to be predicted as a number of larger size. Furthermore, the variances of $f(W_{i,j})$ decrease with sample size growing, which provides a sanity check for our models.

4.3 Learning

Following the previous method, we obtain a matrix $\mathbf{S}'$ consisting of all guesses of $\mathbf{S}_{i,j}$'s.[10] While clear correlations between the guess $\mathbf{S}'$ and $\mathbf{S}$ were observed, the guess was not good enough by itself for the limited number of samples that we used. In the following, we exploit the "absolute-circulant" structure of the secret key to improve our guess. The experimental results described below are based on our script `attack.py`.

[10] We ignore diagonal elements because they are public.

Determining the locations. Notice that all $\mathbf{S}_{i,j}$'s in a same diagonal are of the same absolute value, hence we used a simple trick to enhance the contrast between large and small coefficients. It consists in calculating

$$\mathcal{W}_k = \sum_{i=0}^{n-1} \mathbf{S}'^2_{i,(i+k) \bmod n}$$

as the weight of the k-th diagonal. Since we used two different features for coefficients near/far from the main diagonal, for better comparison, the first $n/2-1$ weights were scaled by their maximum and so were the last $n/2$ weights. We denote by $\mathcal{W}_k^-$ the first $n/2-1$ scaled weights and by $\mathcal{W}_k^+$ the last $n/2$ ones.

As illustrated in Figure 6, the scaled weights of those diagonals consisting of large coefficients are significantly larger than others. A straightforward method to locate large coefficients is to pick the N_b largest scaled weights.

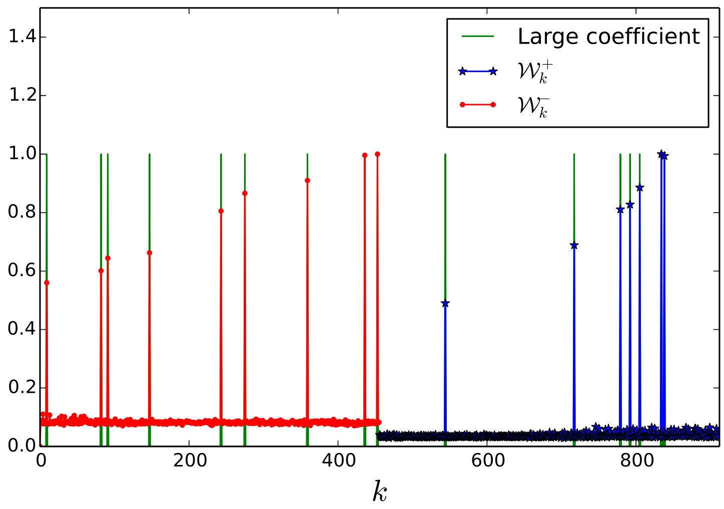

Fig. 6. Large coefficients and scaled weights. Experimental values measure over 400 000 samples.

Verified by experimental results, we were able to perfectly locate all large coefficients, provided we collected sufficient signatures. For different sample size, i.e. the number of signatures, we respectively tested 20 instances and checked the accuracy of locations for large coefficients. All experimental data is illustrated in Table 2.

#signatures	13/16	14/16	15/16	16/16
50 000	5	3	6	6
100 000	-	-	-	20
200 000	-	-	-	20
400 000	-	-	-	20

Table 2. Experimental measure of location accuracy. The column, labeled by $K/16$, shows the number of tested instances in which the largest N_b scaled weights corresponded to exactly K large coefficient diagonals.

Determining the signs. We straightforward assumed the sign of measured feature $f(W_{i,j})$ is the same as that of $\mathbf{S}_{i,j}$, when $\mathbf{S}_{i,j} = \pm b$. Unlike guessing locations, we could not recover all signs of large coefficients exactly, but as the sample size grows, we were still able to get a high accuracy, denoted by p. Then, we may expect to recover all signs of large coefficients in each row exactly with a probability $p_{row} = p^{N_b}$ (in our case $N_b = 16$).

Moreover, we noticed that the accuracy of guessing signs for large coefficients in the lower triangle, i.e. $\mathbf{S}_{i,j}$ with $i > j$, is higher than that for large coefficients in the upper triangle, thus we denote by p_l and p_u the accuracy corresponding to the lower and upper triangle. That may suggest us to guess the signs of large coefficients from the last row to the first row. Table 3 exhibits the experimental data for p_l, p_u, p and p_{row}.

#signs	p_l	p_u	p	p_{row}
400 000	0.9975	0.9939	0.9956	0.9323
200 000	0.9920	0.9731	0.9826	0.7546
100 000	0.9722	0.9330	0.9536	0.4675
50 000	0.9273	0.8589	0.8921	0.1608

Table 3. Experimental measures for p_l, p_u, p and p_{row}. All values measure over 20 instances.

Comparing guessing locations, guessing signs is much more sensitive to the number of signatures. That is because the sign information of $\mathbf{S}_{i,j}$ only comes from $f(W_{i,j})$ rather than all features in the same diagonal so that it requires a more precise measurement. Furthermore, we tried a modified model for guessing signs: in training phase, we mapped $\mathbf{S}_{i,j}$ to $\lfloor \mathbf{S}_{i,j}/b \rceil$ and then find x_ℓ's determining the global feature. Intuitively, the modified model further emphasizes large coefficients, but it performed almost the same as the current model in practice.

5 Exploiting Partial Secret Key Knowledge in Lattice Attacks

Using the technique described in last section, we are able to recover exactly all off-diagonal large coefficients in a row, with high probability (in addition to the diagonal coefficient D). First, we show how to adapt the BDD-uSVP attack, by exploiting the known coefficients of a row $\mathbf{s}_k$ to decrease the distance of the BDD target to the lattice, making the problem easier. Then, we show a more involved version, where we also decrease the dimension of the lattice while maintaining its volume. While not much is gained to recover a first secret row $\mathbf{s}_k$, this technique makes guessing the rest of the key much faster.

In later discussion, assume that we have already successfully determined all $-b, b$ and D coefficients in $\mathbf{s}_k$. Let $M = \{m_0, \cdots, m_{N_b}\}$ be the set of all m's such that $\mathbf{S}_{k,m} \in \{-b, b, D\}$ where $m_0 < \cdots < m_{N_b}$. We still focus on the concrete parameter set $(n, D, b, N_b, N_1) = (912, 912, 28, 16, 432)$.

5.1 Direct BDD-uSVP attack

Let $\mathbf{t} \in \mathbb{Z}^n$ such that, if $|\mathbf{S}_{k,i}| > 1$, $t_i = \mathbf{S}_{k,i}$, otherwise $t_i = 0$, then $\operatorname{dist}(\mathbf{t}, \mathcal{L}) = \sqrt{N_1}$. We construct a new lattice $\mathcal{L}'$ with a basis

$$\mathbf{P}' = \begin{pmatrix} \mathbf{t} & 1 \\ \mathbf{P} & 0 \end{pmatrix} \in \mathbb{Z}^{(n+1)\times(n+1)},$$

we have $\operatorname{vol}(\mathcal{L}') = \operatorname{vol}(\mathcal{L}) \approx D^n$ and $\mathcal{L}'$ contains a vector of Euclidean norm $\sqrt{N_1 + 1} \ll D$. Thus, to recover $\mathbf{s}_k$, it suffices to solve uSVP on $\mathcal{L}'$.

New estimations of the blocksize required by BKZ to solve uSVP were given in [5] and have been confirmed by theoretical analysis and experiments in [3]. Following these results, we claim that $\mathbf{s}_k$ could be recovered by BKZ-β when β satisfies:

$$\sqrt{\frac{\beta}{n+1}} \cdot \sqrt{N_1 + 1} \leq \delta_\beta^{2\beta - n - 1} \cdot D^{\frac{n}{n+1}}.$$

We conclude that running BKZ-β with $\beta = 146$ should be sufficient to break the scheme. Typically [9, 1], it is estimated that BKZ-β converges after about 16 tours,[11] therefore making $16(n+1)$ calls to SVP-β:

$$C_{\text{BKZ-}\beta} = 16(n+1) \cdot C_{\text{SVP-}\beta}.$$

Though the factor 16 may shrink by increasing the blocksize β' progressively from 2 to β. Estimation of the cost of $C_{\text{SVP-}\beta}$ varies a bit in the literature, also depending on the algorithm used. The standard reference for estimating the cost enumeration is [11], which gives a cost of $2^{0.270\beta \ln \beta - 1.019\beta + 16.10}$ [4, 10] clock-cycles. Alternatively, the Gauss-Sieve algorithm [25] with dimension for free and other tricks showed a running time of $2^{0.396\beta + 8.4}$ clock cycles [12].

[11] A theoretical estimate of the expected tour number is $\Omega\left(\frac{n^2}{\beta^2}\log n\right)$ suggested in [19].

Those two methods lead respectively to estimates of 2^{78} and 2^{80} clock-cycles to recover one secret row. One could of course repeat the attack over each row, but below, we present a strategy that slightly reduces the cost of guessing a first row, and greatly reduces the cost of guessing all the other rows.

Remark 3. These numbers are likely to be over-estimates. Indeed, while cost predictions have not been provided, the enumeration algorithms have been sped up in practice recently with the discrete-pruning technique [15, 6, 34]. More importantly, it has seen a significant progress on sieve [12, 2] and the General Sieve Kernel (G6K) [2] has pushed the SVP challenge record up to SVP-157. Unfortunately, the record timing on SVP challenges are difficult to use, as they only solve SVP up to an approximation factor of 1.05, which is significantly easier than the exact SVP typically used in BKZ. Moreover, the BKZ used in G6K proceeds in a different way; its behavior remains to be studied.

5.2 BDD-uSVP attack with dimension reduction

Next we detail how to also reduce the dimension of $\mathcal{L}'$ but maintain its volume, when exploiting known coefficients of a BDD solution.

Let $\mathbf{H} = (h_{i,j})_{i,j}$ be the HNF (Hermite Normal Form) of $\mathbf{P}$ satisfying:

- $h_{i,i} > 0$;
- $h_{j,i} \in \mathbb{Z}_{h_{i,i}}$ for any $j > i$.
- $h_{j,i} = 0$ for any $j < i$.

Let $I = \{i \mid h_{i,i} > 1\}$. In general, $|I|$ is very small (say ≤ 5), for example $|I| = 1$ if $\det(\mathbf{H})$ is square-free. Thus we have, with a high probability, that $I \cap M = \emptyset$, i.e. $h_{m,m} = 1$ for any $m \in M$. If not so, we choose another row $\mathbf{s}_{k'}$ of $\mathbf{S}$. Let $\{l_0, \cdots, l_{n-2-N_b}\} = \mathbb{Z}_n \setminus M$ where $l_0 < \cdots < l_{n-2-N_b}$.

Let $\mathbf{H}' = (h'_{i,j})_{i,j}$ be a matrix of size $(n - N_b - 1) \times (n - N_b - 1)$, in which $h'_{i,j} = h_{l_i,l_j}$. Let $\mathbf{a} = (a_0, \cdots, a_{n-N_b-2})$ where $a_i = \sum_{m \in M} \mathbf{S}_{k,m} h_{m,l_i}$. Let $\mathcal{L}'$ be the lattice generated by

$$\mathbf{B} = \begin{pmatrix} \mathbf{H}' & \\ \mathbf{a} & 1 \end{pmatrix} \in \mathbb{Z}^{(n-N_b)\times(n-N_b)}.$$

We first have that

$$\mathrm{vol}(\mathcal{L}') = \det(\mathbf{H}') = \frac{\det(\mathbf{H})}{\prod_{m \in M} h_{m,m}} = \det(\mathbf{H}) = \mathrm{vol}(\mathcal{L}).$$

Secondly, we can prove that $\mathcal{L}'$ has an unusually short vector corresponding to all small coefficients of $\mathbf{s}_k$. Indeed, let $\mathbf{c} \in \mathbb{Z}^n$ such that $\mathbf{cH} = \mathbf{s}_k$, then $c_m = \mathbf{S}_{k,m}$ for any $m \in M$ thanks to $h_{m,m} = 1$. Let $\mathbf{c}' = (c_{l_0}, \cdots, c_{l_{n-2-N_b}})$, then

$$(\mathbf{c}', 1)\mathbf{B} = (\mathbf{c}'\mathbf{H}' + \mathbf{a}, 1) = (s_{l_0}, \cdots, s_{l_{n-2-N_b}}, 1) := \mathbf{v}'.$$

Notice that $\|\mathbf{v}'\| = \sqrt{N_1 + 1} \ll \mathrm{vol}(\mathcal{L}')^{\frac{1}{n-N_b}} \approx D^{\frac{n}{n-N_b}}$, we may use uSVP oracle to find $\mathbf{v}'$.

Using the same argument as in the previous subsection, we could recover $\mathbf{v}'$, namely $\mathbf{s}_k$, by BKZ-β when β satisfies:

$$\sqrt{\frac{\beta}{n-N_b}} \cdot \sqrt{N_1+1} \leq \delta_\beta^{2\beta-n+N_b} \cdot D^{\frac{n}{n-N_b}}.$$

This condition is satisfied for $\beta = 138$. Based respectively on [11] and [12], this gives attack in 2^{73} and 2^{77} clock-cycles. Again, these numbers should be taken with a grain of salt (see Remark 3).

5.3 Cheaply recovering all the other rows

Once a vector $\mathbf{s}_k$ has been fully recovered, we have much more information on all the other secret rows. In particular, we know all the positions of the 0, and this allows to decrease the dimension from n to $N_b + N_1 + 1$.

As in previous section we are able to construct a (N_b+N_1+1)-dimensional lattice $\mathcal{L}'$ of the same volume as $\mathcal{L}$ and containing a vector of length $\sqrt{N_b \cdot b^2 + N_1 + 1}$. Then, using BKZ-50 is enough[12] to recover the target vector and the cost is negligible compared to the cost of the first step.

6 Cryptanalysis of the DRS v2 Scheme

To resist the aforementioned attack, we did suggest in our conference version [36] some potential countermeasures summarized as follows:

1. *Densely distributed coefficients.* For a sparse yet wide set $\{0, \pm 1, \pm b\}$, the gap between 1 and b allows one to detect large coefficients with much more confidence (see Figures 4). Thus noise coefficients should better spread over an interval of integers $\{-u, \cdots, u\}$.
2. *No "absolute-circulant" structure.* Indeed the "absolute-circulant" structure is exploited to significantly improve the guess (see Figure 6). Additionally, such structure seems unnecessary for the sake of small key size; indeed, the whole matrix, could be streamed by a PRG, only keeping the seed as the new secret key.[13]
3. *Perturbation/drowning.* Depending on the situation, adding well-designed perturbation may [28] or may not [21, 14] be an effective countermeasure against statistical attacks. Drowning is a similar idea in spirit, but the added noise has a fixed distribution, typically much larger than what is to be hidden.

Along with above suggestions, we have also explicitly mentioned that countermeasures 1 and 2 may only mitigate the attack, but would not fully seal the leak.

[12] The required blocksize can be much smaller, but we should use a different estimation for δ_β for small β [11, 35].

[13] Variants can be designed so that each row can be generated on demand.

Following up on our initial work, the DRS team proposed a new variant [30, 33], called the DRS v2 scheme. Essentially, the modifications concern the key generation algorithm, and follow the principles of our countermeasures 1 and 2. Precisely, the modified secret key is still a diagonally dominant matrix

$$\mathbf{S} = D \cdot \mathbf{I}_n - \mathbf{M}$$

but the noise matrix $\mathbf{M}$ is randomly chosen among all possible candidates:

$$\mathbf{M} \leftarrow U\left(\{\mathbf{v} \in \mathbb{Z}^n \mid \|\mathbf{v}\|_1 \leq D\}^n\right).$$

We refer to [30] for detailed algorithms.

The DRS v2 scheme is assumed to withstand learning attack but without detailed security analysis. The impact of learning attack on it remains unclear. In this section, we investigate the effectiveness of our learning attack against DRS v2. In brief, we are not able anymore to recover partial secret coefficients exactly, but we are still able to recover key-related information that is helpful to lower the cost of lattice attacks. Qualitatively, we confirm that a leak is still here. Quantitatively, we estimate a drop by at least 30-bits of security having seen about 2^{30} signatures, for the first set of parameters of DRS v2.

6.1 Leakage

We note that none of the changes introduced in DRS v2 affects the principle of the leak from Section 4.1, and therefore follow essentially the same basic steps. However, because the absolute-circulant structure has been removed, we can not apply our amplification trick anymore, and we therefore require much more data to detect biases. Furthermore, there is no distinct gap between the values to be guessed, contrary to the 'large' coefficients $\pm b$ of the first version of DRS.

It therefore becomes much less practical to carry out our experiments directly on cryptographically large instance. Instead, we proceed with experiments in small dimensions and extrapolate the behaviour.

Features. We just re-use all 13 features introduced in Section 4.2. Let us recall that the models are some linear combinations of these features trained using the least-square fit method.

Model training. We follow the previous idea of endowing each diagonal with a model. A minor difference is that we train two functions f^d and f^o for the main diagonal and others respectively. One may also split f^o into more pieces as done before, but that does not improve the performance finally given the same training data. For each dimension n, we generate 5 random DRS v2 instances for training. And we separately train the model for each sample size N.

Computation. We let the computation run for about a week on 20 cores, for various parameters n ranging from 16 to 512. To reproduce the experiments, one just needs to execute the shell script `full_attack_all.sh` available at the public repository.

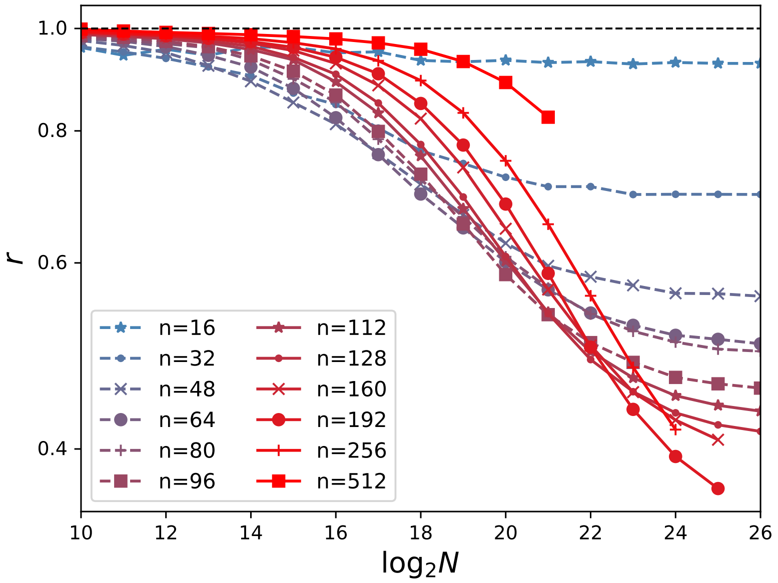

Fig. 7. Experimental measure of the average $r(n, N)$. The y-axis is set as log-scale. The average is taken over 5 instances.

Performance of models. With the model, we make a guess $\mathbf{S}'$ of the secret $\mathbf{S}$ through many signatures. The number of used signatures is equal to the sample size of training the model. We evaluate the performance of the model parametrized by (n, N) with

$$r(n, N) = \frac{1}{n} \sum_{i=0}^{n-1} \frac{\|\mathbf{s}_i - \mathbf{s}_i'\|}{\|\mathbf{s}_i - D\mathbf{e}_i\|}$$

where $\mathbf{e}_i$ is the i-th row of $\mathbf{I}_n$. This is closely related to the complexity of the BDD-uSVP attack (see Section 5): the smaller $r(n, N)$ is, the lower the complexity is; and $D\mathbf{e}_i$ is the default target made without any secret information.

Figure 7 shows the evolution of $r(n, N)$. It seems that given n, $r(n, N)$ decreasingly converges to some $r_\infty(n)$ as N grows. This shows a limitation of our models: the model is incapable of computing the exact secret key no matter how many samples are provided. But on a positive note, $r_\infty(n)$ behaves like a decreasing function of n, thus our model remains asymptotically effective (at the cost of more samples).

Despite some effort, we did not find a satisfactory fit for $r(n, N)$. We therefore only extrapolate an upper bound of $r(n, N)$ based on the apparent monotonicity in n of the derived function $r'(n, N') = r(n, N' \cdot n^2)$ for all N'. Indeed, Figure 8 suggests that $r'(n, N')$ seems monotonically decreasing with n. Since

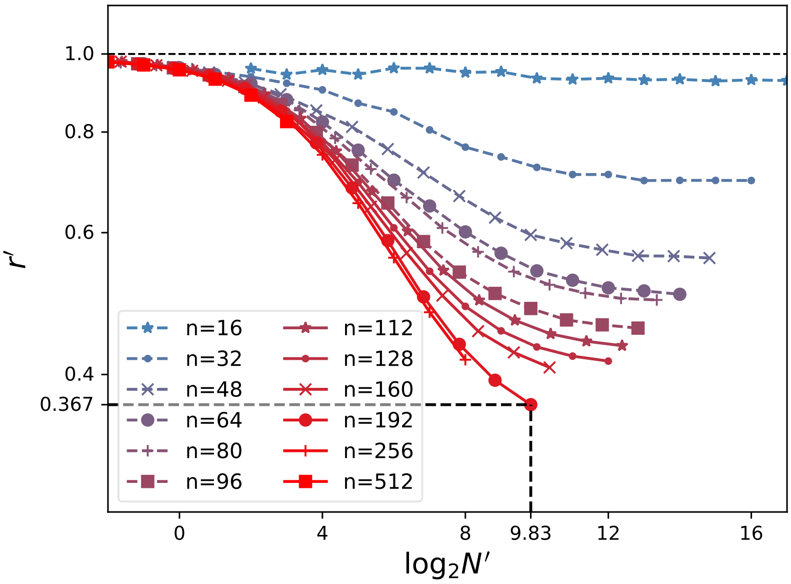

Fig. 8. Experimental measure of $r'(n, N') = r(n, N' \cdot n^2)$. The y-axis is set as log-scale.

$r'(192, 2^{9.83}) \approx 0.367$ is the minimum achieved by experiment, we use it as an estimate of $r'(n, 2^{9.83})$ for $n \geq 192$. Indeed our experiment can be pushed further; more training instances and larger sample sizes certainly refine the estimate. Yet this needs more computation resources and seems unlikely to improve the final security estimate significantly. Qualitatively, our results confirm that the DRS v2 scheme still suffers from a statistical leak.

6.2 Quantitative security analysis

In this subsection, we focus on the concrete parameter set $(n, D) = (1108, 1108)$ claimed to provide at least 128-bits of security in [30].

Security estimate without leaks. The original security estimate of DRS v2 [30, 33] considers the cost of the BDD-uSVP attack following the argument in [16] for which we use the shorthand the *2008 estimate*. But as shown in [3], 2008 estimate is now made obsolete by the *2016 estimate* [5] that we follow in Section 5. For a fair comparison, let us first make a security estimate based on the 2016 estimate. As claimed in Section 5, some secret row vector, say $\mathbf{s}_0$ could be recovered by BKZ-β when β satisfies:

$$\sqrt{\frac{\beta}{n+1}} \cdot \|\mathbf{s}_0 - D\mathbf{e}_0\| \leq \delta_\beta^{2\beta-n-1} \cdot D^{\frac{n}{n+1}}.$$

We suppose $\|\mathbf{s}_0 - D\mathbf{e}_0\| \approx \sqrt{2}\cdot\frac{n-1}{\sqrt{n}}$, which is implicit in the original estimate [30, 33]. Then we verify that $\beta = 220$ satisfies the condition. This gives attack in 2^{126} and 2^{110} clock-cycles based on [11] and [12] respectively.

Refined security estimate with leaks. According to Section 6.1, given the model parametrized by (n, N) along with N signatures, one can compute a guess $\mathbf{S}'$ of the secret basis $\mathbf{S}$ such that $\|\mathbf{s}_0 - \mathbf{s}'_0\| \approx r(n,N)\cdot\|\mathbf{s}_0 - D\mathbf{e}_0\|$. Then the required blocksize β for BKZ suffices to satisfies:

$$\sqrt{\frac{\beta}{n+1}}\cdot\|\mathbf{s}_0 - D\mathbf{e}_0\|\cdot r(n,N) \le \delta_\beta^{2\beta-n-1}\cdot D^{\frac{n}{n+1}}.$$

For a concrete comparison, we set $N \approx 2^{9.83}\cdot n^2 \approx 2^{30.1}$, then $r(n,N) \le 0.367$ as mentioned in Section 6.1. A routine computation yields that one may recover $\mathbf{s}_0$ by BKZ-178 within 2^{98} and 2^{93} clock-cycles according to [11] and [12]. We highlight again that the security estimates are probably over-estimates: the cost of BKZ can be refined hopefully [2] and there may be some non-negligible difference between $r(n,N)$ and its upper bound 0.367. In addition, larger N would further reduce these numbers. Note that $N \approx 2^{30}$ is far from the maximal query number 2^{64} suggested in the NIST call for proposal [27, Sec 4.A.4].

Since we cannot determine some secret coefficients with high confidence, the "dimension reduction" technique (see Section 5.2) does not apply to DRS v2. Additionally, as the absolute-circulant structure is removed, we do not have as in Section 5.3 a much faster way of recovering the other secret vectors.

However, the cost of recovering all the rows should be significantly less than n times that of recovering a first secret row. Indeed, as we accumulate knowledge of many short vectors, the cost of the lattice attack should decrease significantly. Suppose we have recovered i secret vectors, say $\mathbf{s}_0, \cdots, \mathbf{s}_{i-1}$, we now proceed to recover $\mathbf{s}_i$. Let $\mathbf{proj}_i(\mathbf{s})$ denote the projection of $\mathbf{s}$ on $\mathrm{span}(\mathbf{s}_0, \cdots, \mathbf{s}_{i-1})$ and $\mathbf{orth}_i(\mathbf{s}) = \mathbf{s} - \mathbf{proj}_i(\mathbf{s})$. We consider the following lattice $\mathcal{L}'$ with a basis

$$\mathbf{P}' = \begin{pmatrix} \mathbf{orth}_i(\mathbf{s}_i - \mathbf{s}'_i) & 1 \\ \mathbf{orth}_i(\mathbf{s}_i) & 0 \\ \vdots & 0 \\ \mathbf{orth}_i(\mathbf{s}_{n-1}) & 0 \end{pmatrix} \in \mathbb{Q}^{(n+1-i)\times(n+1)}.$$

All $\mathbf{s}_i$'s are nearly orthogonal and $\|\mathbf{s}_i\| \approx D$, hence $\mathrm{vol}(\mathcal{L}') \approx D^{n-i}$. A standard heuristic suggests that $\|\mathbf{orth}_i(\mathbf{s}_i - \mathbf{s}'_i)\| \approx \sqrt{\frac{n+1-i}{n+1}}\|\mathbf{s}_i - \mathbf{s}'_i\|$. Consequently, given $\mathbf{s}_0, \cdots, \mathbf{s}_{i-1}$, one can recover $\mathbf{s}_i$ by BKZ-β with β satisfying:

$$\sqrt{\frac{\beta}{n+1-i}}\cdot\sqrt{\frac{n+1-i}{n+1}}\|\mathbf{s}_i - D\mathbf{e}_i\|\cdot r(n,N) \le \delta_\beta^{2\beta-(n+1-i)}\cdot D^{\frac{n-i}{n+1-i}}.$$

Figure 9 shows the blocksize β required for the recovery of the i-th secret row vector: with more vectors being revealed, β decreases rapidly. The cost of BKZ is (super-)exponential with β, thus the cost of full key recovery is dominated by

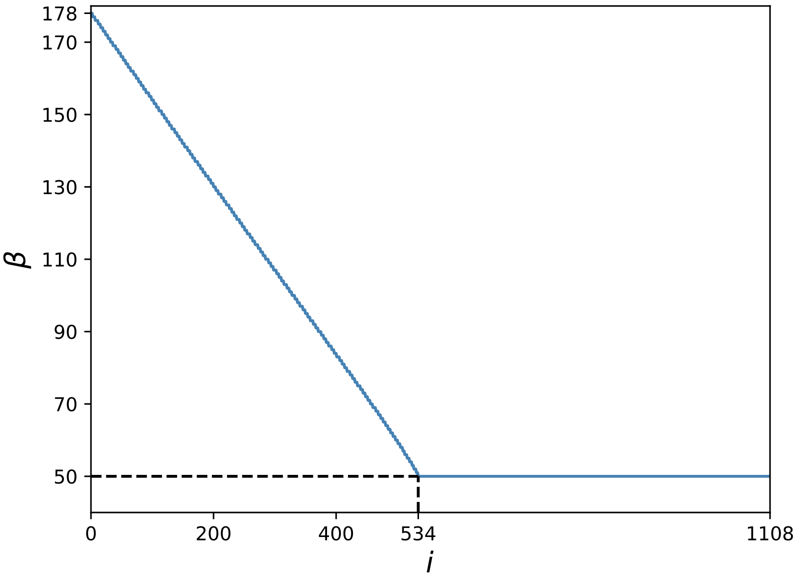

Fig. 9. Required β for recovering $\mathbf{s}_i$.

the first calls of BKZ. Note that we simply set $\beta = 50$ when its prediction is less than 50, which only results in a negligible impact on the security estimate. By some numerical computation, the overall cost is around 2^{101} and 2^{97} clock-cycles based on [11] and [12] respectively, that is, only about 2^4 times the cost of recovering a single vector, and not $n \approx 2^{10}$ times as the naive approach would.

7 Conclusion

We have shown that the DRS scheme is in principle susceptible to a statistical attack: signatures do leak information about the secret key. More concretely, for the first set of parameters submitted to the NIST [29], we have shown its security should be considered below 80-bits after $100\,000 \approx 2^{17}$ signatures have been released, contradicting the original claim of 128-bits of security. While such a large number of signatures may not be released in many applications, it remains much lower than the bound of 2^{64} signatures given by the NIST call for proposal [27, Sec 4.A.4].

We also verify that despite the countermeasure, signature transcripts of the DRS v2 scheme [30, 33] still leak some exploitable information on the secret key. After 2^{30} signatures have been released, the security provided by the first suggested parameter set should be considered below 100-bits rather than the original claim of 128-bits.

In addition, we would like to clarify that our security estimates are likely to be over-estimates. It seems possible to improve the learning step by using more signatures, better-chosen features and advanced machine learning techniques. Working with larger instances will also help to evaluate the model performance more accurately. As for the cost of lattice attacks, we did stick to the best known attack methodology in this paper. But we do not take account of the state-of-the-art algorithms, e.g. discrete pruning technique for enumeration [15, 6, 34] and the General Sieve Kernel for sieve [2], that have unfortunately not yet been the object of easily usable predictions. Overall, it is not clear how much effort an accurate cryptanalysis of DRS deserves. In our view, our current attack suffices to demonstrate the need to fix the leak of all current DRS schemes [29, 30, 33], and maybe to re-parametrize them.

7.1 Potential countermeasure

We note nevertheless that this statistical attack seems much less powerful than the statistical attacks presented in [26, 14] against the original schemes GGH [18] and NTRUSign [21]. Indeed, our attack requires much more signatures, and still only recovers partial secret key information. In this light, we *do not* conclude that the approach of [31, 29, 30, 33] is to be discarded at once, at least if it shows competitive performances.

As we saw (Section 6), the mitigation countermeasures did make the attack less powerful, but did not seal the leak, and we predict a noticeable decrease of security from this leak. Given this track record of non-provable countermeasure to leaks in lattice based signature schemes, we find the formal approaches preferable. Perturbation and drowning could be promising. We note that the problem of directly trying to forge a signature seems harder than recovering the secret key with the current parameters of DRS. This means that allowing larger vectors for signatures (up to a certain cross-over point) should not affect security. This gives a lot of room for perturbation or drowning, for which ad-hoc concrete statistical statements could plausibly be made, maybe exploiting Rényi divergence as in [5, 7].

Finally we insist that a much more thorough analysis of the statistical properties of the scheme should be provided to sustain its security. A statistical argument would be much more reassuring than the experimental failure of the type of attack described in this paper.

References

[1] Albrecht, M.R.: On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL. In: EUROCRYPT 2017. (2017) 103–129

[2] Albrecht, M.R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E.W., Stevens, M.: The General Sieve Kernel and New Records in Lattice Reduction. In: EUROCRYPT 2019. (2019) 717–746

[3] Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the Expected Cost of Solving uSVP and Applications to LWE. In: ASIACRYPT 2017. (2017) 297–322

[4] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of Learning with Errors. Journal of Mathematical Cryptology **9**(3) (2015)

[5] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum Key Exchange—A New Hope. In: USENIX Security 16. (2016) 327–343

[6] Aono, Y., Nguyen, P.Q.: Random Sampling Revisited: Lattice Enumeration with Discrete Pruning. In: EUROCRYPT 2017. (2017) 65–102

[7] Bai, S., Langlois, A., Lepoint, T., Stehlé, D., Steinfeld, R.: Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather Than the Statistical Distance. In: ASIACRYPT 2015, Springer (2015) 3–24

[8] Basak, D., Pal, S., Patranabis, D.C.: Support vector regression. Neural Information Processing-Letters and Reviews **11**(10) (2007) 203–224

[9] Chen, Y.: Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. PhD thesis (2013)

[10] Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better Lattice Security Estimates(full version). http://www.di.ens.fr/∼ychen/research/Full_BKZ.pdf.

[11] Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better Lattice Security Estimates. In: ASIACRYPT 2011. (2011) 1–20

[12] Ducas, L.: Shortest Vector from Lattice Sieving: a Few Dimensions for Free. In: EUROCRYPT 2018. (2018) 125–145

[13] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice Signatures and Bimodal Gaussians. In: CRYPTO 2013. (2013) 40–56

[14] Ducas, L., Nguyen, P.Q.: Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures. In: ASIACRYPT 2012. (2012) 433–450

[15] Fukase, M., Kashiwabara, K.: An accelerated algorithm for solving SVP based on statistical analysis. Journal of Information Processing **23**(1) (2015) 67–80

[16] Gama, N., Nguyen, P.Q.: Predicting Lattice Reduction. In: EUROCRYPT 2008. (2008) 31–51

[17] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. (2008) 197–206

[18] Goldreich, O., Goldwasser, S., Halevi, S.: Public-Key Cryptosystems from Lattice Reduction Problems. In: CRYPTO '97. (1997) 112–131

[19] Hanrot, G., Pujol, X., Stehlé, D.: Analyzing Blockwise Lattice Algorithms Using Dynamical Systems. In: CRYPTO 2011. (2011) 447–464

[20] Haoyu Li, Renzhang Liu, A.N., Pan, Y.: Cryptanalysis of the Randomized Version of a Lattice-Based Signature Scheme from PKC'08. In: ACISP 2018. (2018) to appear

[21] Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSIGN: Digital Signatures Using the NTRU Lattice. In: CT-RSA 2003. (2003) 122–140

[22] Hu, Y., Wang, B., He, W.: NTRUSign With a New Perturbation. IEEE Trans. Information Theory **54**(7) (2008) 3216–3221

[23] Liaw, A., Wiener, M., et al.: Classification and regression by randomForest. R news **2**(3) (2002) 18–22

[24] Lyubashevsky, V.: Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In: ASIACRYPT 2009. (2009) 598–616

[25] Micciancio, D., Voulgaris, P.: Faster exponential time algorithms for the shortest vector problem. In: SODA 2010. (2010) 1468–1480

[26] Nguyen, P.Q., Regev, O.: Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures. In: EUROCRYPT 2006. (2006) 271–288

[27] NIST: Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process (December 2016) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[28] Peikert, C.: An Efficient and Parallel Gaussian Sampler for Lattices. In: CRYPTO 2010. (2010) 80–97

[29] Plantard, T., Sipasseuth, A., Dumondelle, C., Susilo, W.: DRS : Diagonal dominant Reduction for lattice-based Signature. Submitted to the NIST Post-Quantum Cryptography Project https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[30] Plantard, T., Sipasseuth, A., Dumondelle, C., Susilo, W.: DRS : Diagonal dominant Reduction for lattice-based Signature Version 2. https://documents.uow.edu.au/ thomaspl/drs/current/specification.pdf.

[31] Plantard, T., Susilo, W., Win, K.T.: A Digital Signature Scheme Based on CVP_∞. In: PKC 2008. (2008) 288–307

[32] Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming **66**(1-3) (1994) 181–199

[33] Sipasseuth, A., Plantard, T., Susilo, W.: Improving the Security of the DRS Scheme with Uniformly Chosen Random Noise. In: ACISP 2019. (2019) 119–137

[34] Teruya, T., Kashiwabara, K., Hanaoka, G.: Fast Lattice Basis Reduction Suitable for Massive Parallelization and Its Application to the Shortest Vector Problem. In: PKC 2018. 437–460

[35] Yu, Y., Ducas, L.: Second Order Statistical Behavior of LLL and BKZ. In: Selected Areas in Cryptography - SAC 2017. (2017) 3–22

[36] Yu, Y., Ducas, L.: Learning Strikes Again: the Case of the DRS Signature Scheme. In: ASIACRYPT 2018. (2018) 525–543