Learning Strikes Again: the Case of the DRS Signature Scheme

Yang Yu[1] Léo Ducas[2]

[1]Tsinghua University

[2]Centrum Wiskunde & Informatica

January 2019, London

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice
  - Statistical learning $\Rightarrow$ secret key information leaks

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice
  - Statistical learning $\Rightarrow$ secret key information leaks
  - Lattice techniques $\Rightarrow$ better use of leaks

This is a cryptanalysis work...

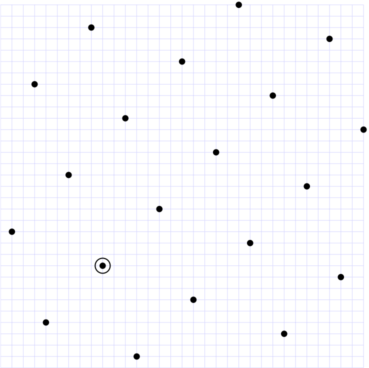
- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice
  - Statistical learning $\Rightarrow$ secret key information leaks
  - Lattice techniques $\Rightarrow$ better use of leaks
- They claim that Parameter Set-I offers **at least 128-bits** of security. We show that it actually offers **at most 80-bits** of security!

Outline

1. Background
2. DRS signature
3. Learning secret key coefficients
4. Exploiting the leaks
5. Countermeasures

Outline

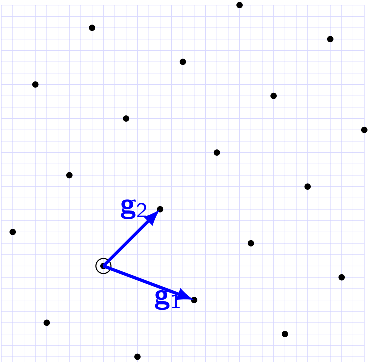
Lattice



Definition

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

Lattice

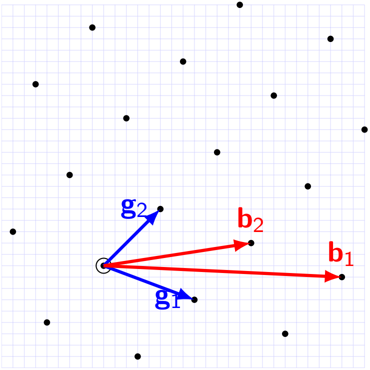


Definition

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

A lattice is generated by its basis $\mathbf{G} = (\mathbf{g}_1, \cdots, \mathbf{g}_n) \in \mathbb{R}^{n \times m}$, e.g. $\mathcal{L} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

Lattice



Definition

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

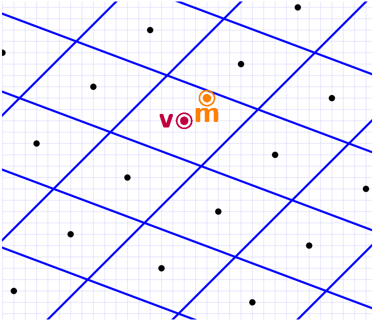
A lattice is generated by its basis $\mathbf{G} = (\mathbf{g}_1, \cdots, \mathbf{g}_n) \in \mathbb{R}^{n \times m}$, e.g. $\mathcal{L} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

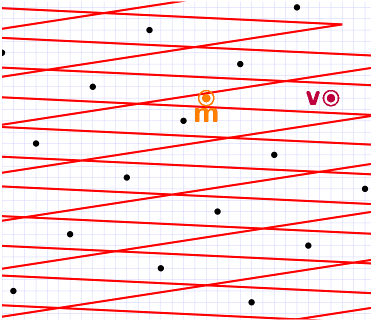
$\mathcal{L}$ has infinitely many bases

$\mathbf{G}$ is good, $\mathbf{B}$ is bad.

Finding close vectors

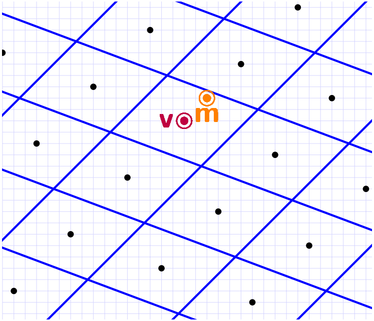
Each basis defines a parallelepiped $\mathcal{P}$.

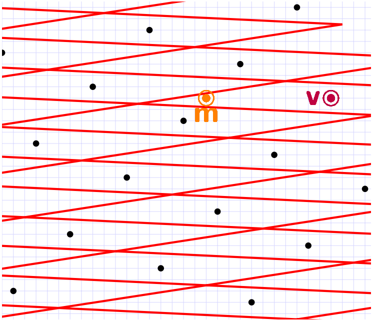




Finding close vectors

Each basis defines a parallelepiped $\mathcal{P}$.





Babai's round-off algorithm outputs $\mathbf{v} \in \mathcal{L}$ such that $\mathbf{v} - \mathbf{m} \in \mathcal{P}$.

GGH & NTRUSign schemes

Public key: $\mathbf{P}$, secret key: $\mathbf{S}$

Sign

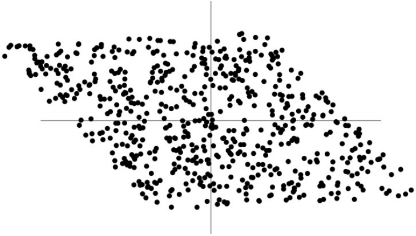
1. Hash the message to a random vector $\mathbf{m}$
2. Round $\mathbf{m}$ (using $\mathbf{S}$) to $\mathbf{v} \in \mathcal{L}$

Verify

1. Check $\mathbf{v} \in \mathcal{L}$ (using $\mathbf{P}$)
2. Check $\mathbf{v}$ is close to $\mathbf{m}$

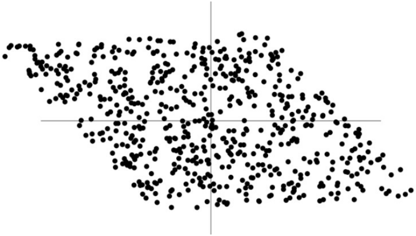
GGH & NTRUSign are insecure!

$\mathbf{v} - \mathbf{m} \in \mathcal{P}(\mathbf{S}) \Rightarrow (\mathbf{v}, \mathbf{m})$ leaks some information of $\mathbf{S}$.



GGH & NTRUSign are insecure!

$\mathbf{v} - \mathbf{m} \in \mathcal{P}(\mathbf{S}) \Rightarrow (\mathbf{v}, \mathbf{m})$ leaks some information of $\mathbf{S}$.



GGH and NTRUSign were broken by “learning the parallelepiped” [NR06].

Some countermeasures were also broken by a similar attack [DN12].

Countermeasures

Let us focus on Hash-then-Sign approach!

Provably secure method [GPV08]

- rounding based on Gaussian sampling
- $\mathbf{v} - \mathbf{m}$ is independent of $\mathbf{S}$

Countermeasures

Let us focus on Hash-then-Sign approach!

Provably secure method [GPV08]

- rounding based on Gaussian sampling
- $\mathbf{v} - \mathbf{m}$ is independent of $\mathbf{S}$

Heuristic method [PSW08]

- rounding based on CVP w.r.t ℓ_∞-norm
- the support of $\mathbf{v} - \mathbf{m}$ is independent of $\mathbf{S}$
- DRS [PSDS17] is an instantiation, submitted to the NIST.

Outline

1. Background
2. DRS signature
3. Learning secret key coefficients
4. Exploiting the leaks
5. Countermeasures

DRS

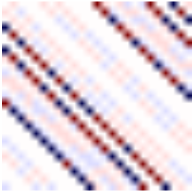
DRS $=$ **D**iagonal-dominant **R**eduction **S**ignature

DRS

DRS = **D**iagonal-dominant **R**eduction **S**ignature

Parameters: (n, D, b, N_b, N_1)

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

$$\mathbf{S} = \begin{pmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{pmatrix} +$$ 

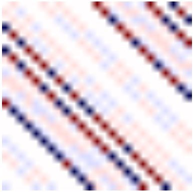
DRS

DRS = **D**iagonal-dominant **R**eduction **S**ignature

Parameters: (n, D, b, N_b, N_1)

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

$$\mathbf{S} = \begin{pmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{pmatrix} +$$



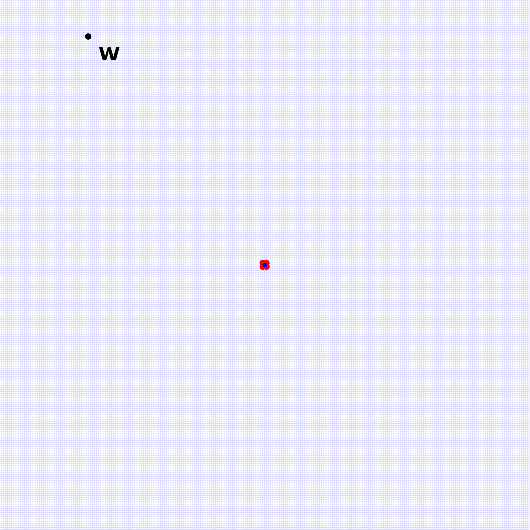
$\leftarrow \begin{cases} D > b \cdot N_b + N_1; \\ \text{absolute circulant} \end{cases}$

Message reduction algorithm

Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$
Output: a reduced message $\mathbf{w}$ such that $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

$\mathbf{w} \leftarrow \mathbf{m}$, $i \leftarrow 0$
repeat
    $\mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor_{\to 0} \cdot \mathbf{s}_i$
    $i \leftarrow (i+1) \bmod n$
until $\|\mathbf{w}\|_\infty < D$
return $\mathbf{w}$

Message reduction algorithm



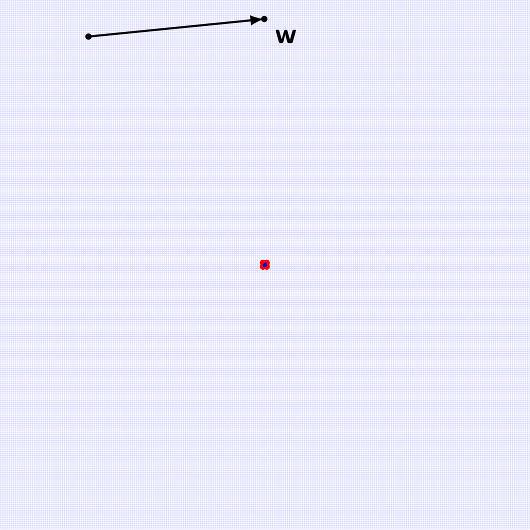
Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$

Output: a reduced message $\mathbf{w}$ s.t. $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

1: $\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor_{\rightarrow 0} \cdot \mathbf{s}_i$
4: $\quad i \leftarrow (i+1) \bmod n$
5: **until** $\|\mathbf{w}\|_\infty < D$
6: return $\mathbf{w}$

$\mathbf{s}_1 = (10, 1), \mathbf{s}_2 = (-1, 10)$
$\mathbf{w} = (-933, 1208)$

Message reduction algorithm

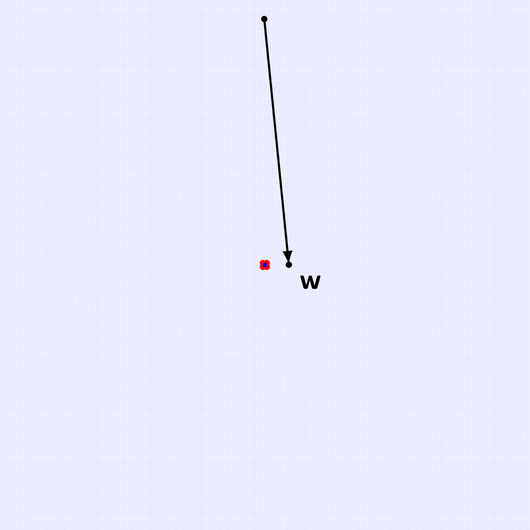


Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$
Output: a reduced message $\mathbf{w}$ s.t. $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

$\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0$
repeat
  $\mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor_{\to 0} \cdot \mathbf{s}_i$
  $i \leftarrow (i+1) \bmod n$
until $\|\mathbf{w}\|_\infty < D$
return $\mathbf{w}$

$\mathbf{s}_1 = (10, 1), \mathbf{s}_2 = (-1, 10)$
$\mathbf{w} = (-933, 1208)$
$\mathbf{w} = \mathbf{w} - (-93) \cdot \mathbf{s}_1 = (-3, 1301)$

Message reduction algorithm



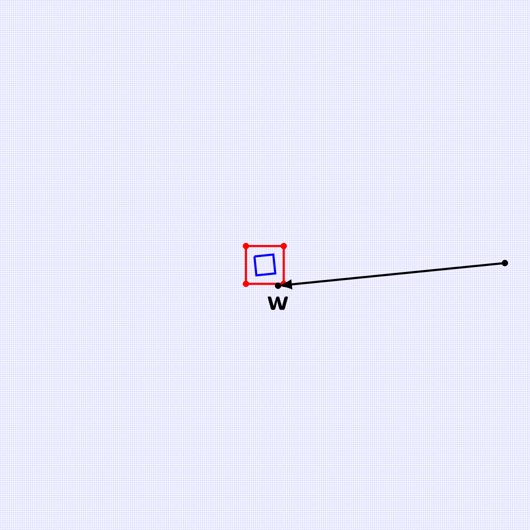
Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$

Output: a reduced message $\mathbf{w}$ s.t. $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

$\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0$
repeat
    $\mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor_{\rightarrow 0} \cdot \mathbf{s}_i$
    $i \leftarrow (i+1) \bmod n$
until $\|\mathbf{w}\|_\infty < D$
return $\mathbf{w}$

$\mathbf{s}_1 = (10, 1), \mathbf{s}_2 = (-1, 10)$
$\mathbf{w} = (-933, 1208)$
$\mathbf{w} = \mathbf{w} - (-93) \cdot \mathbf{s}_1 = (-3, 1301)$
$\mathbf{w} = \mathbf{w} - 130 \cdot \mathbf{s}_2 = (127, 1)$

Message reduction algorithm

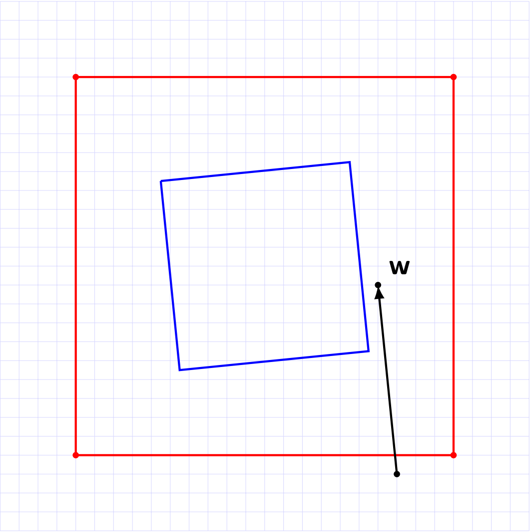


Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$
Output: a reduced message $\mathbf{w}$ s.t. $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

1: $\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor_{\rightarrow 0} \cdot \mathbf{s}_i$
4: $\quad i \leftarrow (i+1) \bmod n$
5: **until** $\|\mathbf{w}\|_\infty < D$
6: return $\mathbf{w}$

$\mathbf{s}_1 = (10, 1), \mathbf{s}_2 = (-1, 10)$
$\mathbf{w} = (-933, 1208)$
$\mathbf{w} = \mathbf{w} - (-93) \cdot \mathbf{s}_1 = (-3, 1301)$
$\mathbf{w} = \mathbf{w} - 130 \cdot \mathbf{s}_2 = (127, 1)$
$\mathbf{w} = \mathbf{w} - 12 \cdot \mathbf{s}_1 = (7, -11)$

Message reduction algorithm



Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$

Output: a reduced message $\mathbf{w}$ s.t. $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

```
w ← m, i ← 0
repeat
    w ← w − ⌊w_i / D⌉_{→0} · s_i
    i ← (i + 1) mod n
until ‖w‖_∞ < D
return w
```

$\mathbf{s}_1 = (10, 1), \mathbf{s}_2 = (-1, 10)$

$\mathbf{w} = (-933, 1208)$

$\mathbf{w} = \mathbf{w} - (-93) \cdot \mathbf{s}_1 = (-3, 1301)$

$\mathbf{w} = \mathbf{w} - 130 \cdot \mathbf{s}_2 = (127, 1)$

$\mathbf{w} = \mathbf{w} - 12 \cdot \mathbf{s}_1 = (7, -11)$

$\mathbf{w} = \mathbf{w} - (-1) \cdot \mathbf{s}_2 = (6, -1)$

Message reduction algorithm

Intuition: use $\mathbf{s}_i$ to reduce

- w_i decreases a lot
- for $j \neq i$, w_j increases a bit

Message reduction algorithm

Intuition: use $\mathbf{s}_i$ to reduce

- w_i decreases a lot
- for $j \neq i$, w_j increases a bit

A reduction at i : $\mathbf{w} \to \mathbf{w} - q\mathbf{s}_i$, $q = \lfloor \frac{w_i}{D} \rfloor_{\to 0}$

$$\begin{aligned}
\|\mathbf{w} - q\mathbf{s}_i\|_1 &= \sum_{k \neq i} |w_k - qs_{i,k}| + |w_i| - |q| \cdot D \qquad (q \cdot w_i > 0) \\
&\leq \sum_{k \neq i} (|w_k| + |qs_{i,k}|) + |w_i| - |q| \cdot D \\
&= \|\mathbf{w}\|_1 - |q| \cdot (D - \sum_{k \neq i} |s_{i,k}|) \\
&< \|\mathbf{w}\|_1 \qquad \text{(diagonal dominance)}
\end{aligned}$$

Message reduction algorithm

Intuition: use $\mathbf{s}_i$ to reduce

- w_i decreases a lot
- for $j \neq i$, w_j increases a bit

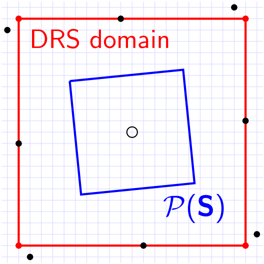
A reduction at i : $\mathbf{w} \to \mathbf{w} - q\mathbf{s}_i$, $q = \lfloor \frac{w_i}{D} \rceil_{\to 0}$

$$
\begin{aligned}
\|\mathbf{w} - q\mathbf{s}_i\|_1 &= \sum_{k \neq i} |w_k - q s_{i,k}| + |w_i| - |q| \cdot D \qquad (q \cdot w_i > 0) \\
&\leq \sum_{k \neq i} (|w_k| + |q s_{i,k}|) + |w_i| - |q| \cdot D \\
&= \|\mathbf{w}\|_1 - |q| \cdot (D - \sum_{k \neq i} |s_{i,k}|) \\
&< \|\mathbf{w}\|_1 \qquad \text{(diagonal dominance)}
\end{aligned}
$$

$\Rightarrow$ message reduction always terminates!

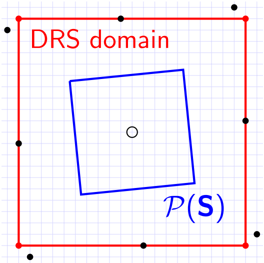
Resistance to NR attack

The support of $\mathbf{w}$: $(-D, D)^n$



Resistance to NR attack

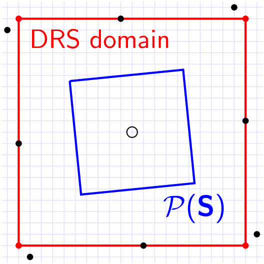
The support of $\mathbf{w}$: $(-D, D)^n$



The support is “zero-knowledge”

Resistance to NR attack

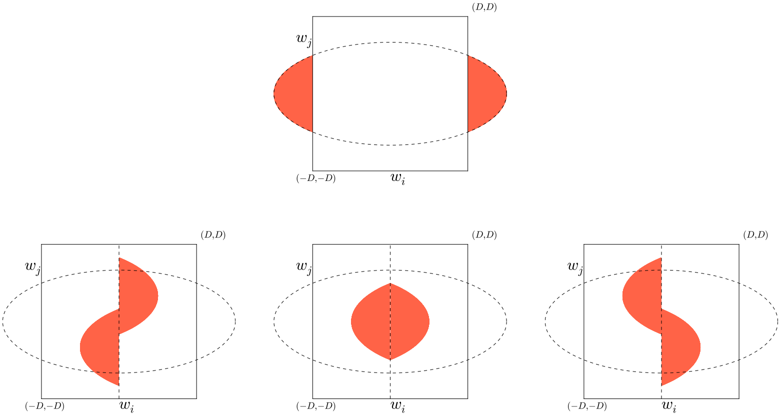
The support of $\mathbf{w}$: $(-D, D)^n$



The support is "zero-knowledge", but **maybe the distribution is not!**

Outline

Intuition



$\mathbf{S}_{i,j} = -b$ $\qquad$ $\mathbf{S}_{i,j} = 0$ $\qquad$ $\mathbf{S}_{i,j} = b$

Correlations

Two sources of correlations between (w_i, w_j)

- reduction at i and $\mathbf{S}_{i,j} \neq 0$

Correlations

Two sources of correlations between (w_i, w_j)

- reduction at i and $\mathbf{S}_{i,j} \neq 0$
- reduction at k and $\mathbf{S}_{k,i}, \mathbf{S}_{k,j} \neq 0$

Correlations

Two sources of correlations between (w_i, w_j)

- reduction at i and $\mathbf{S}_{i,j} \neq 0$ ★
- reduction at k and $\mathbf{S}_{k,i}, \mathbf{S}_{k,j} \neq 0$

Correlations

Two sources of correlations between (w_i, w_j)

- reduction at i and $\mathbf{S}_{i,j} \neq 0$ ★
- reduction at k and $\mathbf{S}_{k,i}, \mathbf{S}_{k,j} \neq 0$

$\Rightarrow$ $\mathbf{S}_{i,j}$ should be **strongly related** to $W_{i,j}$ (the distribution of (w_i, w_j)) !

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$?

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

- cascading phenomenon: a reduction triggers another one.
- parasite correlations

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

- cascading phenomenon: a reduction triggers another one.
- parasite correlations

$\Rightarrow$ **Search for the best linear fit f ?**

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

- cascading phenomenon: a reduction triggers another one.
- parasite correlations

$\Rightarrow$ **Search for the best linear fit f ?**

Search space for all linear f: too large!

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

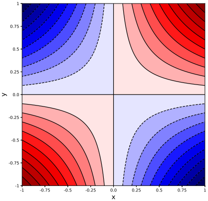
- cascading phenomenon: a reduction triggers another one.
- parasite correlations

$\Rightarrow$ **Search for the best linear fit f ?**

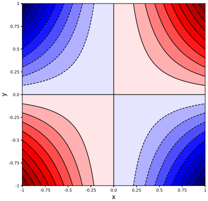
Search space for all linear f: too large!
$\Rightarrow$ **choose some features $\{f_i\}$ and search in $\mathrm{span}(\{f_i\})$, i.e. $f = \sum x_\ell f_\ell$**

Lower degree moments:

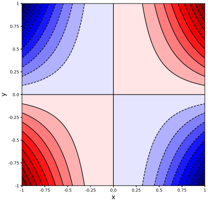
$f_1(W) = \mathbb{E}(w_i w_j)$



$f_2(W) = \mathbb{E}(w_i \cdot |w_i|^{1/2} \cdot w_j)$

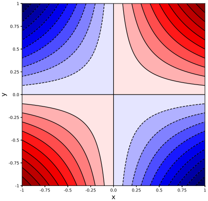


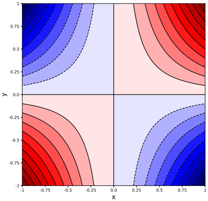
$f_3(W) = \mathbb{E}(w_i \cdot |w_i| \cdot w_j)$

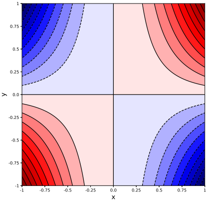


Lower degree moments:

$f_1(W) = \mathbb{E}(w_i w_j)$ $\qquad$ $f_2(W) = \mathbb{E}(w_i \cdot |w_i|^{1/2} \cdot w_j)$ $\qquad$ $f_3(W) = \mathbb{E}(w_i \cdot |w_i| \cdot w_j)$

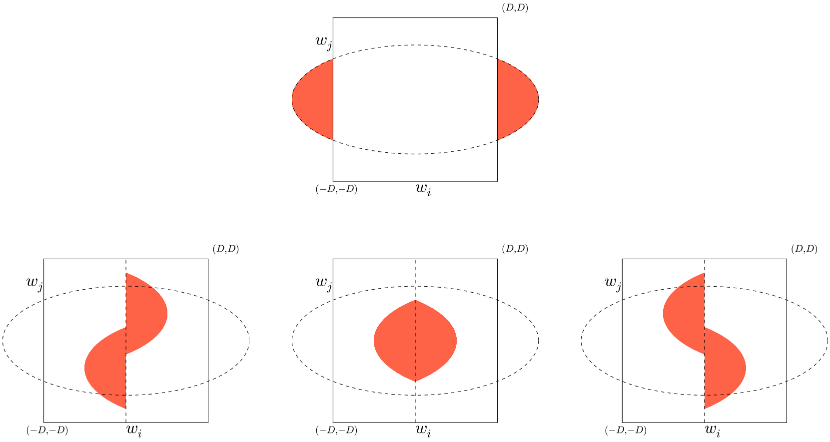






Not enough!

Training — feature selection

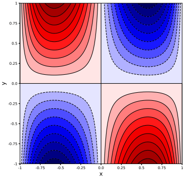


$\mathbf{S}_{i,j} = -b$ $\mathbf{S}_{i,j} = 0$ $\mathbf{S}_{i,j} = b$

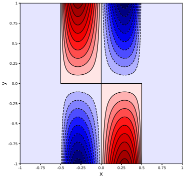
Training — feature selection

Pay more attention to the central region (i.e. $|w_i|$ small).

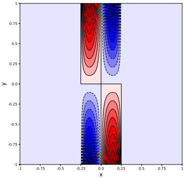
$$f_4 = \mathbb{E}(w_i(w_i-1)(w_i+1)w_j)$$



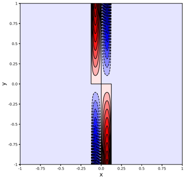
$$f_5 = \mathbb{E}(2w_i(2w_i-1)(2w_i+1)w_j \mid |2w_i| \leq 1)$$



$$f_6 = \mathbb{E}(4w_i(4w_i-1)(4w_i+1)w_j \mid |4w_i| \leq 1)$$



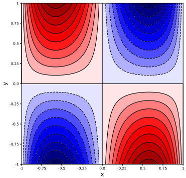
$$f_7 = \mathbb{E}(8w_i(8w_i-1)(8w_i+1)w_j \mid |8w_i| \leq 1)$$



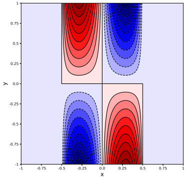
Training — feature selection

Pay more attention to the central region (i.e. $|w_i|$ small).

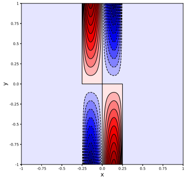
$$f_4 = \mathbb{E}(w_i(w_i-1)(w_i+1)w_j)$$



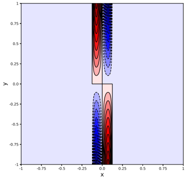
$$f_5 = \mathbb{E}(2w_i(2w_i-1)(2w_i+1)w_j \mid |2w_i| \le 1)$$



$$f_6 = \mathbb{E}(4w_i(4w_i-1)(4w_i+1)w_j \mid |4w_i| \le 1)$$



$$f_7 = \mathbb{E}(8w_i(8w_i-1)(8w_i+1)w_j \mid |8w_i| \le 1)$$



Together with transposes (i.e. $f^t(w_i, w_j) = f(w_j, w_i)$), we finally selected $7 \times 2 - 1 = 13$ features in experiments.

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

- $f^+ = \sum x^+ f_\ell, f^- = \sum x^- f_\ell$ according to $(i - j \bmod n)$.

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

- $f^+ = \sum x^+ f_\ell, f^- = \sum x^- f_\ell$ according to $(i - j \bmod n)$.

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

- $f^+ = \sum x^+ f_\ell, f^- = \sum x^- f_\ell$ according to $(i - j \bmod n)$.

Build models by **least-square fit** method

- 30 instances and 400 000 samples per instances
- 38 core-hours

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

- $f^+ = \sum x^+ f_\ell, f^- = \sum x^- f_\ell$ according to $(i - j \bmod n)$.

Build models by **least-square fit** method

- 30 instances and 400 000 samples per instances
- 38 core-hours

Possible improvements

- advanced machine learning techniques

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

- $f^+ = \sum x^+ f_\ell, f^- = \sum x^- f_\ell$ according to $(i - j \bmod n)$.

Build models by **least-square fit** method

- 30 instances and 400 000 samples per instances
- 38 core-hours

Possible improvements

- advanced machine learning techniques
- more blocks

Training — model construction

$\mathbf{S}_{i,j}$ seems easier to learn when $(i - j \bmod n)$ is smaller.

- $f^+ = \sum x^+ f_\ell, f^- = \sum x^- f_\ell$ according to $(i - j \bmod n)$.

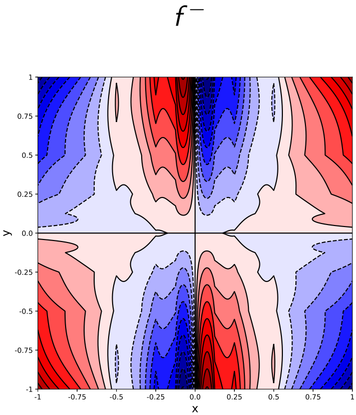
Build models by **least-square fit** method

- 30 instances and 400 000 samples per instances
- 38 core-hours

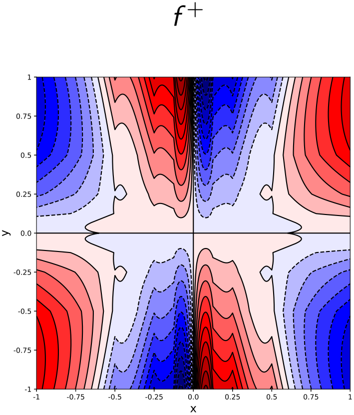
Possible improvements

- advanced machine learning techniques
- more blocks
- new features

The models



f^-
1
0.75
0.5
0.25
0.0
-0.25
-0.5
-0.75
-1
y
-1
-0.75
-0.5
-0.25
0.0
0.25
0.5
0.75
1
x

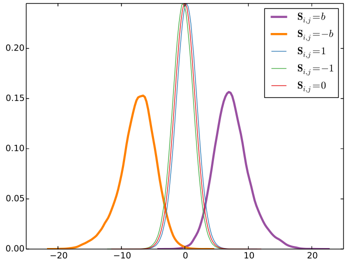


f^+
1
0.75
0.5
0.25
0.0
-0.25
-0.5
-0.75
-1
y
-1
-0.75
-0.5
-0.25
0.0
0.25
0.5
0.75
1
x

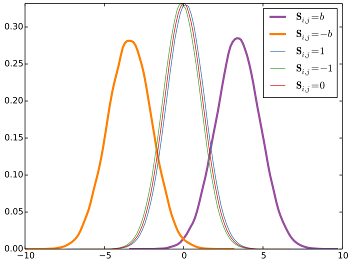
Learning

Let's learn a new $\mathbf{S}$ as $\mathbf{S}' = f(W)$!

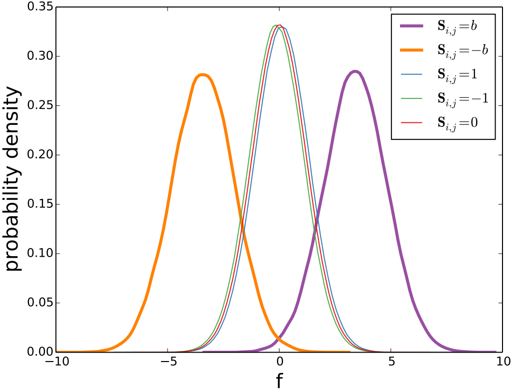
f^-



f^+

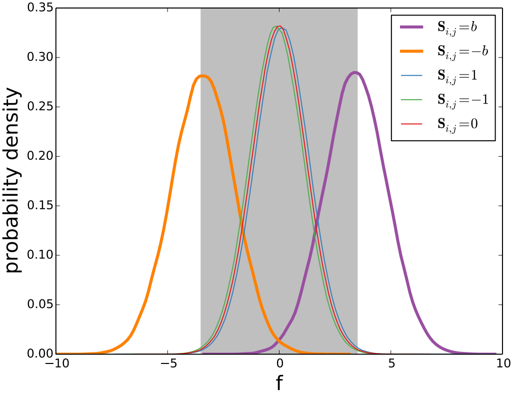


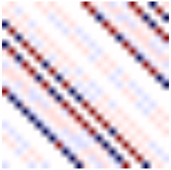
Let's learn a new $\mathbf{S}$ as $\mathbf{S}' = f(W)$!



Learning

Let's learn a new **S** as $\mathbf{S}' = f(W)$!

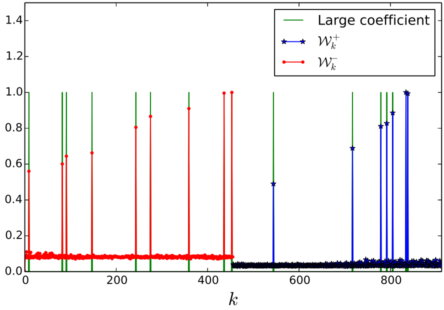


$\mathbf{S} = D \cdot I +$  is "absolute circulant"

$\Rightarrow$ more confidence via diagonal amplification

Learning — location

The weight of k-th diagonal $\mathcal{W}_k = \sum \mathbf{S}'^2_{i,i+k}$



Learning — location

#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: 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.

Learning — location

#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: 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.

We locate all large coefficients successfully!

Learning — location

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

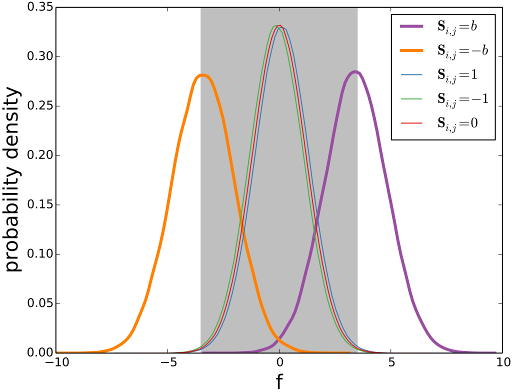
Table: Location accuracy. The column, labeled by $K/16$, shows the number of tested instances in which the largest N_b scaled weights corresponded to exactly K large coefficient diagonals.

We locate all large coefficients successfully!

but we are still missing the signs!

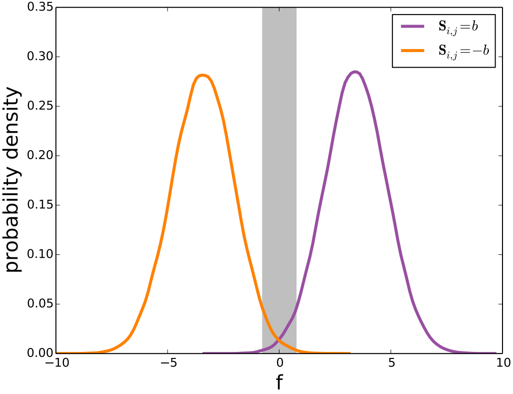
Learning — sign

$\mathbf{S}_{i,j} \in \{\pm b, \pm 1, 0\}$



Learning — sign

$\mathbf{S}_{i,j} \in \{\pm b\}$



Learning — sign

#signatures	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: Experimental measures for p_l, p_u, p and p_{row}.

$p =$ accuracy of guessing the sign of a large coefficient

$p_l =$ accuracy for a large coefficient in the lower triangle

$p_u =$ accuracy for a large coefficient in the upper triangle

$p_{row} = p^{N_b}$

Learning — sign

#signatures	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: Experimental measures for p_l, p_u, p and p_{row}.

$p =$ accuracy of guessing the sign of a large coefficient

$p_l =$ accuracy for a large coefficient in the lower triangle

$p_u =$ accuracy for a large coefficient in the upper triangle

$p_{row} = p^{N_b}$

We can determine all large coefficients in one row!

Learning — sign

#signatures	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: Experimental measures for p_l, p_u, p and p_{row}.

$p =$ accuracy of guessing the sign of a large coefficient

$p_l =$ accuracy for a large coefficient in the lower triangle

$p_u =$ accuracy for a large coefficient in the upper triangle

$p_{row} = p^{N_b}$

We can determine all large coefficients in one row!
However, it is still hard to learn small coefficients...

Outline

BDD & uSVP

BDD (Bounded Distance Decoding)

Given a lattice $\mathcal{L}$ and a target $\mathbf{t}$ "very close" to $\mathcal{L}$, to find $\mathbf{v} \in \mathcal{L}$ minimizing $\|\mathbf{v} - \mathbf{t}\|$.

uSVP (Unique SVP)

Given a lattice $\mathcal{L}$ with $\lambda_1(\mathcal{L}) \ll \lambda_2(\mathcal{L})$, to find its shortest non-zero vector.

BDD & uSVP

BDD (Bounded Distance Decoding)

Given a lattice $\mathcal{L}$ and a target $\mathbf{t}$ "very close" to $\mathcal{L}$, to find $\mathbf{v} \in \mathcal{L}$ minimizing $\|\mathbf{v} - \mathbf{t}\|$.

uSVP (Unique SVP)

Given a lattice $\mathcal{L}$ with $\lambda_1(\mathcal{L}) \ll \lambda_2(\mathcal{L})$, to find its shortest non-zero vector.

BDD $\Rightarrow$ uSVP on $\mathcal{L}'$ spanned by $\begin{pmatrix} \mathbf{B} & \\ \mathbf{t} & 1 \end{pmatrix}$

BDD & uSVP

BDD (Bounded Distance Decoding)

Given a lattice $\mathcal{L}$ and a target $\mathbf{t}$ "very close" to $\mathcal{L}$, to find $\mathbf{v} \in \mathcal{L}$ minimizing $\|\mathbf{v} - \mathbf{t}\|$.

uSVP (Unique SVP)

Given a lattice $\mathcal{L}$ with $\lambda_1(\mathcal{L}) \ll \lambda_2(\mathcal{L})$, to find its shortest non-zero vector.

BDD $\Rightarrow$ uSVP on $\mathcal{L}'$ spanned by $\begin{pmatrix} \mathbf{B} & \\ \mathbf{t} & 1 \end{pmatrix}$

- $\lambda_1(\mathcal{L}') = \sqrt{1 + \mathrm{dist}(\mathbf{t}, \mathcal{L})^2}$
- $\mathrm{vol}(\mathcal{L}') = \mathrm{vol}(\mathcal{L})$

Solving uSVP by BKZ

Required blocksize β

[ADPS16, AGVW17]: $\sqrt{\beta/d} \cdot \lambda_1(\mathcal{L}') \leq \delta_\beta^{2\beta-d} \cdot \mathrm{vol}(\mathcal{L}')^{\frac{1}{d}}$

where $d = \dim(\mathcal{L}')$, $\delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$ $(\beta > 50)$.

Solving uSVP by BKZ

Required blocksize β

[ADPS16, AGVW17]: $\sqrt{\beta/d} \cdot \lambda_1(\mathcal{L}') \leq \delta_\beta^{2\beta-d} \cdot \mathrm{vol}(\mathcal{L}')^{\frac{1}{d}}$

where $d = \dim(\mathcal{L}')$, $\delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$ $(\beta > 50)$.

Cost of BKZ-β

[Che13, Alb17]: $C_{\text{BKZ-}\beta} = 16d \cdot C_{\text{SVP-}\beta}$

Solving uSVP by BKZ

Required blocksize β

[ADPS16, AGVW17]: $\sqrt{\beta/d} \cdot \lambda_1(\mathcal{L}') \leq \delta_\beta^{2\beta-d} \cdot \mathrm{vol}(\mathcal{L}')^{\frac{1}{d}}$

where $d = \dim(\mathcal{L}')$, $\delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e} \right)^{\frac{1}{2(\beta-1)}}$ $(\beta > 50)$.

Cost of BKZ-β

[Che13, Alb17]: $C_{\text{BKZ-}\beta} = 16d \cdot C_{\text{SVP-}\beta}$

Cost of solving SVP-β

- Enum[APS15]: $2^{0.270\beta \ln\beta - 1.019\beta + 16.10}$

Solving uSVP by BKZ

Required blocksize β

[ADPS16, AGVW17]: $\sqrt{\beta/d} \cdot \lambda_1(\mathcal{L}') \leq \delta_\beta^{2\beta-d} \cdot \mathrm{vol}(\mathcal{L}')^{\frac{1}{d}}$

where $d = \dim(\mathcal{L}')$, $\delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$ $(\beta > 50)$.

Cost of BKZ-β

[Che13, Alb17]: $C_{\text{BKZ-}\beta} = 16d \cdot C_{\text{SVP-}\beta}$

Cost of solving SVP-β

- Enum[APS15]: $2^{0.270\beta\ln\beta - 1.019\beta + 16.10}$
- Sieve [Duc17]: $2^{0.396\beta + 8.4}$

Solving uSVP by BKZ

Required blocksize β

[ADPS16, AGVW17]: $\sqrt{\beta/d} \cdot \lambda_1(\mathcal{L}') \leq \delta_\beta^{2\beta-d} \cdot \mathrm{vol}(\mathcal{L}')^{\frac{1}{d}}$

where $d = \dim(\mathcal{L}')$, $\delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$ $(\beta > 50)$.

Cost of BKZ-β

[Che13, Alb17]: $C_{\text{BKZ-}\beta} = 16d \cdot C_{\text{SVP-}\beta}$

Cost of solving SVP-β

- Enum[APS15]: $2^{0.270\beta\ln\beta - 1.019\beta + 16.10}$ ★
- Sieve [Duc17]: $2^{0.396\beta + 8.4}$

Leaks help a lot!

Attack without leaks

- $d = n + 1$, $\lambda_1(\mathcal{L}') = \sqrt{b^2 \cdot N_b + N_1 + 1}$
- cost: $> 2^{128}$

Leaks help a lot!

Attack without leaks

- $d = n + 1$, $\lambda_1(\mathcal{L}') = \sqrt{\mathbf{b^2 \cdot N_b + N_1 + 1}}$
- cost: $> 2^{128}$

Naive attack with leaks

- $d = n + 1$, $\lambda_1(\mathcal{L}') = \sqrt{\mathbf{N_1 + 1}}$
- cost: 2^{78}

Leaks help a lot!

Attack without leaks

- $d = n + 1,\ \lambda_1(\mathcal{L}') = \sqrt{b^2 \cdot N_b + N_1 + 1}$
- cost: $> 2^{128}$

Naive attack with leaks

- $\mathbf{d = n + 1},\ \lambda_1(\mathcal{L}') = \sqrt{N_1 + 1}$
- cost: 2^{78}

Improved attack with leaks

- $\mathbf{d = n - N_b},\ \lambda_1(\mathcal{L}') = \sqrt{N_1 + 1}$
- cost: 2^{73}

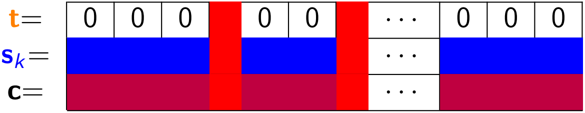
Improved BDD-uSVP attack

Red: $D, \pm b$ **(known)**, **Blue:** $0, \pm 1$ **(unknown)**

$\mathbf{t}=$	0	0	0		0	0		$\cdots$	0	0	0
$\mathbf{s}_k=$								$\cdots$			

Improved BDD-uSVP attack

Let $\mathbf{H} = \begin{pmatrix} * & & & & \\ * & * & & & \\ * & * & 1 & & \\ \vdots & \vdots & & \ddots & \\ * & * & & & 1 \end{pmatrix}$ be $HNF(\mathcal{L})$, and $\mathbf{s} = \mathbf{c}\mathbf{H}$



Improved BDD-uSVP attack

Let $\mathbf{H} = \begin{pmatrix} * & & & & \\ * & * & & & \\ * & * & 1 & & \\ \vdots & \vdots & & \ddots & \\ * & * & & & 1 \end{pmatrix}$ be $HNF(\mathcal{L})$, and $\mathbf{s} = \mathbf{c}\mathbf{H}$

$\mathbf{t}=$	0	0	0		0	0		$\cdots$	0	0	0
$\mathbf{s}_k=$								$\cdots$			
$\mathbf{c}=$								$\cdots$			

Let $\mathbf{M}$ such that

$\mathbf{tM}=$	0	0	$\cdots$	0		$= (\mathbf{0}, \mathbf{r})$
$\mathbf{s}_k\mathbf{M}=$						$= (\mathbf{b}, \mathbf{r})$
$\mathbf{cM}=$						$= (\mathbf{p}, \mathbf{r})$

Improved BDD-uSVP attack

Let $\mathbf{M}^t\mathbf{H}\mathbf{M} = \begin{pmatrix} \mathbf{H}' & \\ \mathbf{H}'' & \mathbf{I} \end{pmatrix}$ and

let $\mathcal{L}'$ be the lattice spanned by $\begin{pmatrix} \mathbf{H}' & \\ \mathbf{t}' = \mathbf{r}\mathbf{H}'' & 1 \end{pmatrix}$

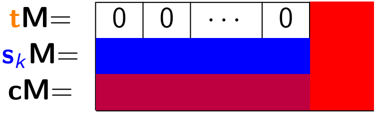
Improved BDD-uSVP attack

Let $\mathbf{M}^t\mathbf{H}\mathbf{M} = \begin{pmatrix} \mathbf{H}' & \\ \mathbf{H}'' & \mathbf{I} \end{pmatrix}$ and

let $\mathcal{L}'$ be the lattice spanned by $\begin{pmatrix} \mathbf{H}' & \\ \mathbf{t}' = \mathbf{r}\mathbf{H}'' & 1 \end{pmatrix}$

- $\dim(\mathcal{L}') = n - N_b$
- $\mathrm{vol}(\mathcal{L}') = \mathrm{vol}(\mathcal{L})$
- $\lambda_1(\mathcal{L}') = \|(\mathbf{b}, 1)\| = \sqrt{N_1 + 1}$

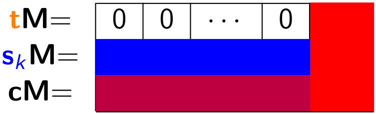
Once one $\mathbf{s}_i$ is recovered exactly $\Rightarrow$ all 0's in $\mathbf{S}$ are determined

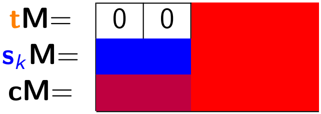


$\dim = n - N_b$

Improved BDD-uSVP attack

Once one $\mathbf{s}_i$ is recovered exactly $\Rightarrow$ all 0's in $\mathbf{S}$ are determined



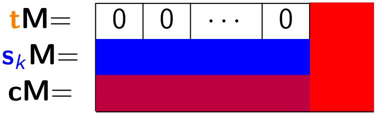


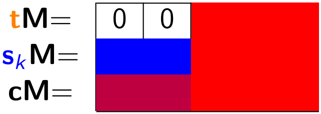
$\dim = n - N_b$

$\dim = N_1 + N_b + 1 \approx n/2$

Improved BDD-uSVP attack

Once one $\mathbf{s}_i$ is recovered exactly $\Rightarrow$ all 0's in $\mathbf{S}$ are determined





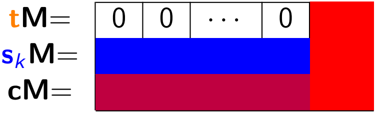
$\dim = n - N_b$

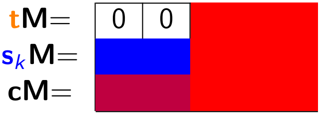
$\dim = N_1 + N_b + 1 \approx n/2$

Recovering secret matrix $\approx$ recovering a first secret.

Improved BDD-uSVP attack

Once one $\mathbf{s}_i$ is recovered exactly $\Rightarrow$ all 0's in $\mathbf{S}$ are determined





dim $= n - N_b$ dim $= N_1 + N_b + 1 \approx n/2$

Recovering secret matrix $\approx$ recovering a first secret.

Can we do better with the help of many $\mathbf{t}_k$ close to $\mathbf{s}_k$? [KF17]

Conclusion

We present a statistical attack against DRS:

- given 100 000 signatures, security is below 80-bits;
- even less with the current progress of lattice algorithms.

Outline

Modified DRS

In DRS: $\mathbf{S} = D \cdot \mathbf{I} + \mathbf{E}$ is diagonal-dominant

Version 1 [PSDS17]

- absolute circulant, $\mathbf{E}_{i,i} = 0$
- three types of coefficients $(\{0\}, \{\pm 1\}, \{\pm b\})$ with fixed numbers

Modified DRS

In DRS: $\mathbf{S} = D \cdot \mathbf{I} + \mathbf{E}$ is diagonal-dominant

Version 1 [PSDS17]

- absolute circulant, $\mathbf{E}_{i,i} = 0$
- three types of coefficients $(\{0\}, \{\pm 1\}, \{\pm b\})$ with fixed numbers

Version 2 [PSDS18]

- $\mathbf{e}_1, \cdots, \mathbf{e}_n \xleftarrow{\$} \{\mathbf{v} \mid \|\mathbf{v}\|_1 < D\}$
- variable diagonal elements

Modified DRS

In DRS: $\mathbf{S} = D \cdot \mathbf{I} + \mathbf{E}$ is diagonal-dominant

Version 1 [PSDS17]

- absolute circulant, $\mathbf{E}_{i,i} = 0$
- three types of coefficients $(\{0\}, \{\pm 1\}, \{\pm b\})$ with fixed numbers

Version 2 [PSDS18]

- $\mathbf{e}_1, \cdots, \mathbf{e}_n \xleftarrow{\$} \{\mathbf{v} \mid \|\mathbf{v}\|_1 < D\}$
- variable diagonal elements

Impact

- no circulant structure $\Rightarrow$ diagonal amplification doesn't work

Modified DRS

In DRS: $\mathbf{S} = D \cdot \mathbf{I} + \mathbf{E}$ is diagonal-dominant

Version 1 [PSDS17]

- absolute circulant, $\mathbf{E}_{i,i} = 0$
- three types of coefficients $(\{0\}, \{\pm 1\}, \{\pm b\})$ with fixed numbers

Version 2 [PSDS18]

- $\mathbf{e}_1, \cdots, \mathbf{e}_n \xleftarrow{\$} \{\mathbf{v} \mid \|\mathbf{v}\|_1 < D\}$
- variable diagonal elements

Impact

- no circulant structure $\Rightarrow$ diagonal amplification doesn't work
- coefficients are less sparsely distributed $\Rightarrow$ less confidence of guessing

Learning attack on modified DRS

We regard $\mathbf{S}_{i,j}$ as a random variable following the same distribution.
Let $\mathbf{S}'$ be the guess of $\mathbf{S}$ and N be the sample size.

Learning attack on modified DRS

We regard $\mathbf{S}_{i,j}$ as a random variable following the same distribution.
Let $\mathbf{S}'$ be the guess of $\mathbf{S}$ and N be the sample size.

As N grows, we hope

- $\mathbf{Var}(\mathbf{S}_{i,j} - \mathbf{S}'_{i,j}) < \mathbf{Var}(\mathbf{S}_{i,j}) \Rightarrow$ more confidence of guessing
- $\|\mathbf{s}_i - \mathbf{s}'_i\| < \|\mathbf{s}_i\| \Rightarrow$ guessing vector gets closer to the lattice

Some experiments on modified DRS

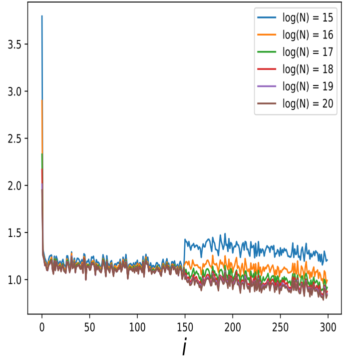
We conducted some experiments on reduced parameters.

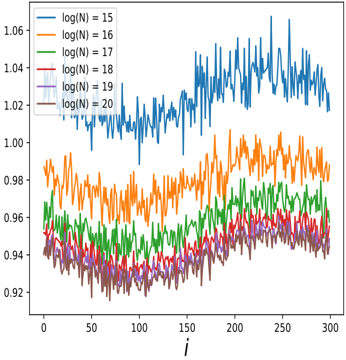
Some experiments on modified DRS

We conducted some experiments on reduced parameters.

We re-used the same approach with same features.

$$\frac{\mathbf{Var}(\mathbf{S}_{j,i+j} - \mathbf{S}'_{j,i+j})}{\mathbf{Var}(\mathbf{S}_{j,i+j})} \qquad \frac{\|\mathbf{s}_i - \mathbf{s}'_i\|}{\|\mathbf{s}_i\|}$$

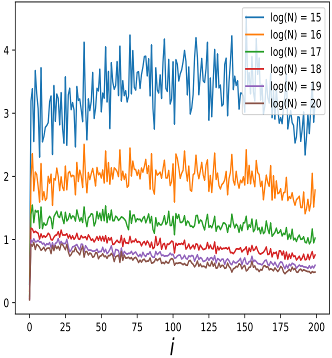




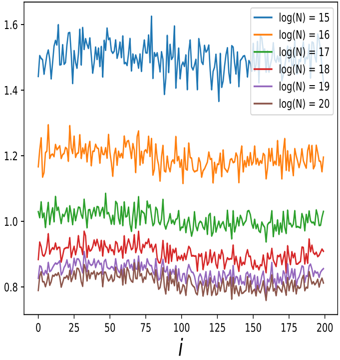
Some experiments on modified DRS

We also tried the case of n blocks and some new features.

$$\frac{\mathbf{Var}(\mathbf{S}_{j,i+j} - \mathbf{S}'_{j,i+j})}{\mathbf{Var}(\mathbf{S}_{j,i+j})}$$

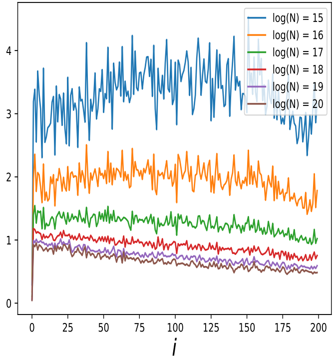


$$\frac{\|\mathbf{s}_i - \mathbf{s}'_i\|}{\|\mathbf{s}_i\|}$$

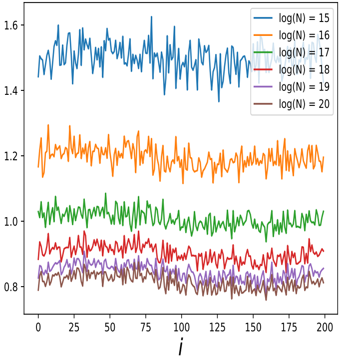


We also tried the case of n blocks and some new features.

$$\frac{\mathbf{Var}(\mathbf{S}_{j,i+j} - \mathbf{S}'_{j,i+j})}{\mathbf{Var}(\mathbf{S}_{j,i+j})}$$



$$\frac{\|\mathbf{s}_i - \mathbf{s}'_i\|}{\|\mathbf{s}_i\|}$$



Further study is ongoing...

Conclusion

A leak still exists despite the new countermeasure.

Conclusion

A leak still exists despite the new countermeasure.

Work in progress

- use **timing leakage** to locate the endpoint of message reduction, then to classify samples and to choose most relevant ones

Open question

- well-designed perturbation & statistical arguments

Thank you!

References

[NR06]. Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures.
Phong Q. Nguyen and Oded Regev. EUROCRYPT 2006.

[DN12]. Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures.
Léo Ducas and Phong Q. Nguyen. ASIACRYPT 2012.

[GPV08]. Trapdoors for hard lattices and new cryptographic constructions.
Craig Gentry and Chris Peikert and Vinod Vaikuntanathan. STOC 2008.

[PSW08]. A Digital Signature Scheme Based on CVP_∞.
Thomas Plantard and Willy Susilo and Khin Than Win. PKC 2008.

[PSDS17]. DRS : Diagonal dominant Reduction for lattice-based Signature.
Thomas Plantard and Arnaud Sipasseuth and Cedric Dumondelle and Willy Susilo. Submitted to the NIST PQC Competition.

[ADPS16]. Post-quantum Key Exchange—A New Hope.
Erdem Alkim and Léo Ducas and Thomas Pöppelmann and Peter Schwabe. USENIX Security 2016.

[AGVW17]. Revisiting the Expected Cost of Solving uSVP and Applications to LWE.
Martin R. Albrecht and Florian Göpfert and Fernando Virdia and Thomas Wunderer. ASIACRYPT 2017.

[Che13]. Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe.
Yuanmi Chen. https://www.theses.fr/2013PA077242.

[Alb17]. On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL.
Martin R. Albrecht. EUROCRYPT 2017.

[APS15]. On the concrete hardness of Learning with Errors.
Martin R. Albrecht and Rachel Player and Sam Scott. Journal of Mathematical Cryptology.

[Duc17]. Shortest Vector from Lattice Sieving: a Few Dimensions for Free.
Léo Ducas. EUROCRYPT 2018.

[KF17]. Revisiting Lattice Attacks on overstretched NTRU parameters.
Paul Kirchner and Pierre-Alain Fouque. EUROCRYPT 2017.

[PSDS18]. DRS : Diagonal dominant Reduction for lattice-based Signature Version 2.
Thomas Plantard and Arnaud Sipasseuth and Cedric Dumondelle and Willy Susilo.
https://www.uow.edu.au/ thomaspl/drs/current/specification.pdf.