Learning Strikes Again: the Case of the DRS Signature Scheme

Yang Yu¹  Léo Ducas²

¹Tsinghua University
²Centrum Wiskunde & Informatica

Asiacrypt 2018
Brisbane, Australia
This is a cryptanalysis work...

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

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!
This is a cryptanalysis work...

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

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!
This is a cryptanalysis work...

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

- **Techniques:** learning & lattice
  - Statistical learning $\implies$ secret key information 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!
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
Outline

1. Background
2. DRS signature
3. Learning secret key coefficients
4. Exploiting the leaks
A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$. A lattice is generated by its basis $G = (g_1, \ldots, g_n) \in \mathbb{R}^n \times \mathbb{R}^m$, e.g. $\mathcal{L} = \{ xG \mid x \in \mathbb{Z}^n \}$. L has infinitely many bases $G$ is good, $B$ is bad.
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. $\mathcal{L} = \{x\mathbf{G} \mid x \in \mathbb{Z}^n\}$. 
A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

A lattice is generated by its basis $G = (g_1, \ldots, g_n) \in \mathbb{R}^{n \times m}$, e.g.

$\mathcal{L} = \{xG \mid x \in \mathbb{Z}^n\}$.

$\mathcal{L}$ has infinitely many bases $G$ is good, $B$ is bad.
Finding Close Vectors

Each basis defines a parallelepiped $P$. 

Babai's round-off algorithm outputs $v \in L$ such that $v - m \in P$. 
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: \( P \), secret key: \( S \)

Sign

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

Verify

1. Check \( v \in \mathcal{L} \) (using \( P \))
2. Check \( v \) is close to \( m \)
GGH & NTRUSign are insecure!

\[ \mathbf{v} - \mathbf{m} \in \mathcal{P}(\mathbf{S}) \Rightarrow (\mathbf{v}, \mathbf{m}) \text{ leaks some information of } \mathbf{S}. \]
GGH & NTRUSign are insecure!

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

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
- $v - m$ is independent of $S$
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 $\ell_\infty$-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
DRS

\[
\text{DRS} = \text{Diagonal-dominant Reduction Signature}
\]
DRS

DRS = Diagonal-dominant Reduction Signature

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

\[
S = \begin{pmatrix}
D & & \\
& D & \\
& & \ddots \\
& & & D
\end{pmatrix} + \text{"absolute circulant"}
\]
DRS = Diagonal-dominant Reduction Signature

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

\[
S = \begin{pmatrix}
D & & \\
& D & \\
& & \ddots \\
& & & D
\end{pmatrix} + \text{“absolute circulant”}
\]
Message reduction algorithm

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

1: $w \leftarrow m$, $i \leftarrow 0$
2: repeat
3: $w \leftarrow w - \left\lfloor \frac{w_i}{D} \right\rfloor \cdot s_i$
4: $i \leftarrow (i + 1) \mod n$
5: until $\|w\|_\infty < D$
6: return $w$
Message reduction algorithm

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

**Output:** a reduced message $w$ such that $w - m \in \mathcal{L}$ and $\|w\|_\infty < D$

1. $w \leftarrow m$, $i \leftarrow 0$
2. repeat
   3. $w \leftarrow w - \left\lfloor \frac{w_i}{D} \right\rfloor \cdot s_i$
   4. $i \leftarrow (i + 1) \bmod n$
5. until $\|w\|_\infty < D$
6. return $w$

**Intuition:** use $s_i$ to reduce

- $w_i$ decreases a lot
- for $j \neq i$, $w_j$ increases a bit
- $\|w\|_1$ is reduced $\Rightarrow$ reduction always terminates!
Resistance to NR attack

The support of $w$: $(-D, D)^n$
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

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

\[ S_{i,j} = -b \] \hspace{2cm} \[ S_{i,j} = 0 \] \hspace{2cm} \[ S_{i,j} = b \]
Can we devise a formula $S_{i,j} \approx f(W_{i,j})$?
Figure out the model

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

- cascading phenomenon: a reduction triggers another one.
- other parasite correlations
Figure out the model

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

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

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

Can we devise a formula $S_{i,j} \approx f(W_{i,j})$? Seems complicated!
- cascading phenomenon: a reduction triggers another one.
- other parasite correlations

⇒ **Search for the best linear fit $f$?**

Search space for all linear $f$: too large!
Figure out the model

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

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

⇒ **Search for the best linear fit $f$?**

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

Lower degree moments:

\[ f_1(W) = \mathbb{E}(w_i w_j) \]
\[ f_2(W) = \mathbb{E}(w_i \cdot |w_i|^{1/2} \cdot w_j) \]
\[ f_3(W) = \mathbb{E}(w_i \cdot |w_i| \cdot w_j) \]
Training — feature selection

Lower degree moments:

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

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

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

Not enough!
Training — feature selection

\[ w_i, w_j \in (-D, -D) \]

\[ S_{i,j} = -b \]

\[ S_{i,j} = 0 \]

\[ S_{i,j} = b \]
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).

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

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

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

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

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.
The model

\[ f = \sum x_\ell f_\ell \]
Let’s learn a new $\mathbf{S}$ as $\mathbf{S}' = f(\mathbf{W})$!
Let’s learn a new $S$ as $S' = f(W)$!
$S = D \cdot I+$ is “absolute circulant”

$\Rightarrow$ more confidence via diagonal amplification
Learning

\[ S = D \cdot I + \text{ is “absolute circulant”} \]

⇒ more confidence via diagonal amplification

- focus on absolute values and put guesses in a same diagonal together
Learning

\[ S = D \cdot I + \] is “absolute circulant”

⇒ more confidence via diagonal amplification
  - focus on absolute values and put guesses in a same diagonal together

**We locate all large coefficients successfully!**
Learning

\[ S = D \cdot I + \] is “absolute circulant”

⇒ more confidence via diagonal amplification

- focus on absolute values and put guesses in a same diagonal together

We locate all large coefficients successfully!
but we are still missing the signs!
Learning

\( S_{i,j} \in \{ \pm b, \pm 1, 0 \} \)

We can determine all large coefficients in one row!

However, it is still hard to learn small coefficients...
Learning

\[ S_{i,j} \in \{ \pm b \} \]
Learning

\[ S_{i,j} \in \{ \pm b \} \]

We can determine all large coefficients in one row!
$S_{i,j} \in \{ \pm b \}$

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

1. Background
2. DRS signature
3. Learning secret key coefficients
4. Exploiting the leaks
Leaks help a lot!

**Attack without leaks**

- \( \text{dim} = n + 1 \), short vector of length \( \sqrt{b^2 \cdot N_b + N_1 + 1} \)
- \( \text{cost: } > 2^{128} \)
Leaks help a lot!

**Attack without leaks**
- \( \text{dim} = n + 1 \), short vector of length \( \sqrt{b^2 \cdot N_\text{b} + N_1 + 1} \)
- cost: \( > 2^{128} \)

**Naive attack with leaks**
- \( \text{dim} = n + 1 \), short vector of length \( \sqrt{N_1 + 1} \)
- cost: \( 2^{78} \)
Leaks help a lot!

<table>
<thead>
<tr>
<th>Attack without leaks</th>
<th>dim = n + 1, short vector of length $\sqrt{b^2 \cdot N_b + N_1 + 1}$</th>
<th>cost: $&gt; 2^{128}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive attack with leaks</td>
<td>dim = n + 1, short vector of length $\sqrt{N_1 + 1}$</td>
<td>cost: $2^{78}$</td>
</tr>
<tr>
<td>Improved attack with leaks</td>
<td>dim = n − $N_b$, short vector of length $\sqrt{N_1 + 1}$</td>
<td>cost: $2^{73}$</td>
</tr>
</tbody>
</table>
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.
Thank you!