Fault Attacks Against Lattice-Based Signatures

T. Espitau[†] P-A. Fouque B. Gérard M. Tibouchi

[†]Lip6, Sorbonne Universités, Paris

August 12, 2016
SAC – 16

Towards postquantum cryptography

- Quantum computers *would* break all currently deployed public-key crypto: RSA, discrete logs, elliptic curves

- Agencies warnings
  - NSA deprecating Suite B (elliptic curves)
  - NIST starting postquantum competition

Towards postquantum cryptography

- In theory, plenty of schemes quantum-resistant
  - Code-based, hash trees, multivariate crypto, isogenies...
  - Almost everything possible with lattices
- In practice, very few actual implementations
  - Secure parameters often unclear
  - Concrete software/hardware implementation papers quite rare
  - Almost no consideration for implementation attacks
- Serious issue for practical postquantum crypto

Implementations of lattice-based schemes (I)

- Implementation of lattice-based crypto:

  Limited and mostly academic

- One scheme has "industry" backing and quite a bit of code: NTRU
  - NTRUEncrypt, ANSI standard, believed to be okay
  - NTRUSign is a trainwreck that has been patched and broken

Implementations of lattice-based schemes (II)

- In terms of practical schemes, other than NTRU, main efforts on signatures
  - GLP: improvement of Lyubashevsky signatures, efficient in SW and HW (CHES'12)
  - BLISS: improvement of GLP, even better (CRYPTO'13, CHES'14)
  - GPV: obtained as part of Ducas, Lyubashevsky, Prest NTRU-based IBE (AC'14),
  - PASSSign (ACNS'14), TESLA (LATINCRYPT 14),...

Implementation attacks vs provable security

Break a provably secure cryptographic scheme:

Solve a hard computational problem

$$\neq$$

Break an implementation

Potentially bypass security proof

"Problem Exists Between Keyboard And Chair"

Implementation attacks

- Side-channel attacks: *Passive physical attacks, exploiting information leakage*
  - Timing attacks, power analysis, EM attacks, cache attacks, acoustic attacks...
- Fault attacks: *Active physical attacks, extract secret information by tampering with the device to cause errors*
  - Faults on memory: lasers, x-rays...
  - Faults on computation: variations in supply voltage, external clock, temperature...

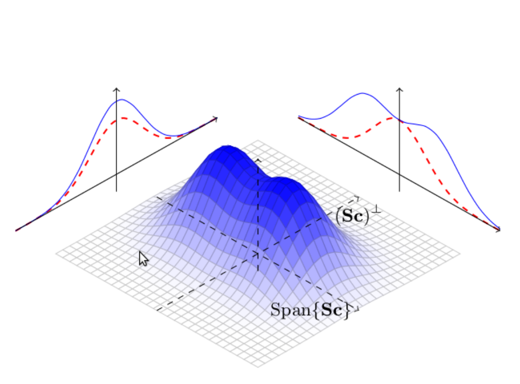
BLISS: the basics

- Introduced by *Ducas, Durmus, Lepoint and Lyubashevsky* at CRYPTO'13
- Improvement of Ring-SIS-based scheme of Lyubashevsky
- Still kind of "Fiat–Shamir signatures"

BLISS: the basics

- Defined over $\mathcal{R} = \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$

- Main improvement: Reduce the size of parameters by Bimodal Gaussian distributions



BLISS: the basics

- Defined over $\mathcal{R} = \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$

- Main improvement: Reduce the size of parameters by Bimodal Gaussian distributions

Distributio Camelus bactrianus

BLISS: key generation

function $\textsc{KeyGen}()$

  choose $\mathbf{f}, \mathbf{g}$ as uniform polynomials with exactly $d_1 = \lceil \delta_1 n \rceil$ entries in $\{\pm 1\}$ and $d_2 = \lceil \delta_2 n \rceil$ entries in $\{\pm 2\}$

  $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)^T \leftarrow (\mathbf{f}, 2\mathbf{g} + 1)^T$

  if $N_\kappa(\mathbf{S}) \geqslant C^2 \cdot 5 \cdot (\lceil \delta_1 n \rceil + 4 \lceil \delta_2 n \rceil) \cdot \kappa$ **then restart**

  if $\mathbf{f}$ is not invertible **then restart**

  $\mathbf{a}_q = (2\mathbf{g} + 1)/\mathbf{f} \bmod q$

  return $(pk = \mathbf{A}, sk = \mathbf{S})$ where $\mathbf{A} = (\mathbf{a}_1 = 2\mathbf{a}_q, q - 2) \bmod 2q$

end function

BLISS: signature

function $\textsc{Sign}(\mu, pk = \mathbf{A}, sk = \mathbf{S})$

  $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ ▷ Gaussian sampling

  $\mathbf{u} = \zeta \cdot \mathbf{a}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$ ▷ $\zeta = 1/(q-2)$

  $\mathbf{c} \leftarrow H(\lfloor \mathbf{u} \rceil_d \bmod p, \mu)$ ▷ special hashing

  choose a random bit b

  $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$

  $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$

  continue with probability $1/\left(M \exp(-\|\mathbf{Sc}\|/(2\sigma^2)\right) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle / \sigma^2)$ otherwise **restart**

  $\mathbf{z}_2^\dagger \leftarrow (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z}_2 \rceil_d) \bmod p$

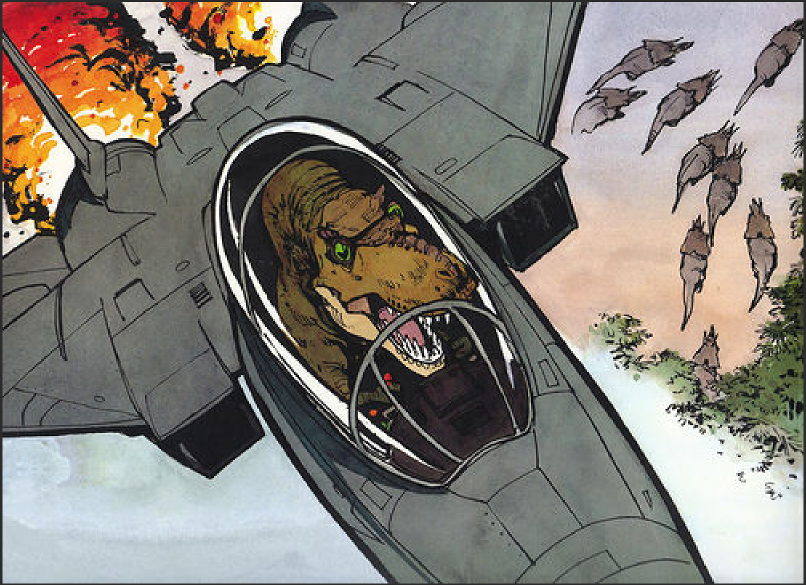
  return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$

end function

BLISS: verification

function $\textsc{Verify}(\mu, \mathbf{A}, (\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c}))$
    if $\|(\mathbf{z}_1 | 2^d \cdot \mathbf{z}_2^\dagger)\|_2 > B_2$ **then** reject
    if $\|(\mathbf{z}_1 | 2^d \cdot \mathbf{z}_2^\dagger)\|_\infty > B_\infty$ **then** reject
    accept iff $\mathbf{c} = H(\lfloor \zeta \cdot \mathbf{a}_1 \cdot \mathbf{z}_1 + \zeta \cdot q \cdot \mathbf{c} \rceil_d + \mathbf{z}_2^\dagger \bmod p, \mu)$
end function

Let's break things!



BLISS: signature

function $\textsc{Sign}(\mu, pk = \mathbf{A}, sk = \mathbf{S})$
$\quad \mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ ▷ Gaussian sampling
$\quad \mathbf{u} = \zeta \cdot \mathbf{a}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$ ▷ $\zeta = 1/(q-2)$
$\quad \mathbf{c} \leftarrow H(\lfloor \mathbf{u} \rceil_d \bmod p, \mu)$ ▷ special hashing
$\quad$ choose a random bit b
$\quad \mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
$\quad \mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
$\quad$ **continue** with probability $1/(M \exp(-\|\mathbf{Sc}\|/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle / \sigma^2)$ otherwise **restart**
$\quad \mathbf{z}_2^\dagger \leftarrow (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z}_2 \rceil_d) \bmod p$
$\quad$ **return** $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function

BLISS: signature

function $\textsc{Sign}(\mu, pk = \mathbf{A}, sk = \mathbf{S})$
  $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ ▷ Gaussian sampling
  $\mathbf{u} = \zeta \cdot \mathbf{a}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$ ▷ $\zeta = 1/(q-2)$
  $\mathbf{c} \leftarrow H(\lfloor \mathbf{u} \rceil_d \bmod p, \mu)$ ▷ special hashing
  choose a random bit b
  $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
  $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
  continue with probability $1/\left(M \exp(-\|\mathbf{Sc}\|/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle / \sigma^2)\right)$ otherwise **restart**
  $\mathbf{z}_2^\dagger \leftarrow (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z}_2 \rceil_d) \bmod p$
  return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function

BLISS: signature

function $\textsc{Sign}(\mu, pk = \mathbf{A},\ sk = \mathbf{S})$

  $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ ▷ Gaussian sampling

  $\mathbf{u} = \zeta \cdot \mathbf{a}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$ ▷ $\zeta = 1/(q-2)$

  $\mathbf{c} \leftarrow H(\lfloor \mathbf{u} \rceil_d \bmod p, \mu)$ ▷ special hashing

  choose a random bit b

  $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$

  $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$

  continue with probability $1/\left(M \exp(-\|\mathbf{Sc}\|/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle / \sigma^2\right)$ otherwise **restart**

  $\mathbf{z}_2^\dagger \leftarrow (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z}_2 \rceil_d) \bmod p$

  return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$

end function

Attacking $\mathbf{y}$

- $\mathbf{y}_1$ ($\equiv$discrete Gaussian) $\approx$ additive mask in

$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod q$$

- Sampling: coefficient by coefficient

- Use fault injection to abort the sampling early $\implies$ faulty signature with a low-degree $\mathbf{y}_1$

- Done by attacking:
  - Branching test of the loop (voltage spike, clock variation...)
  - Contents of the loop counter (lasers, x-rays...)

Attacking $\mathbf{y}$

- $\mathbf{y}_1$ (≡discrete Gaussian) ≈ additive mask in

$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod{q}$$

- Sampling: coefficient by coefficient
- Use fault injection to abort the sampling early $\Longrightarrow$ faulty signature with a low-degree $\mathbf{y}_1$
- Done by attacking:
  - Branching test of the loop (voltage spike, clock variation...)
  - Contents of the loop counter (lasers, x-rays...)

Attacking $\mathbf{y}$

- $\mathbf{y}_1$ (≡discrete Gaussian) ≈ additive mask in

$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod{q}$$

- Sampling: coefficient by coefficient

- Use fault injection to abort the sampling early $\Longrightarrow$ faulty signature with a low-degree $\mathbf{y}_1$

- Done by attacking:
  - Branching test of the loop (voltage spike, clock variation...)
  - Contents of the loop counter (lasers, x-rays...)

Attacking $\mathbf{y}$

- $\mathbf{y}_1$ ($\equiv$discrete Gaussian) $\approx$ additive mask in

$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod{q}$$

- Sampling: coefficient by coefficient

- Use fault injection to abort the sampling early $\Longrightarrow$ faulty signature with a low-degree $\mathbf{y}_1$

- Done by attacking:
  - Branching test of the loop (voltage spike, clock variation...)
  - Contents of the loop counter (lasers, x-rays...)

Attack details (I)

- Signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ invertible (probability $(1-1/q)^n \approx 96\%$):

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod{q}$$

WLOG, $b = 0$ (equivalent keys)

- $\mathbf{s}_1$ is short $\Longrightarrow$ $\mathbf{v}$ very close to lattice

$$L = \mathrm{Span}(q\mathbb{Z}^n, (\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i)_{0 \leqslant i \leqslant m-1})$$

- $\dim(L) = n$ too large to apply lattice reduction

Same relation on subset of coefficients: REDUCE THE DIM

Attack details (I)

- Signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ invertible (probability $(1 - 1/q)^n \approx 96\%$):

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b\mathbf{s}_1 \pmod{q}$$

WLOG, $b = 0$ (equivalent keys)

- $\mathbf{s}_1$ is short $\implies$ $\mathbf{v}$ very close to lattice

$$L = \mathrm{Span}(q\mathbb{Z}^n, (\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i)_{0 \leqslant i \leqslant m-1})$$

- $\dim(L) = n$ too large to apply lattice reduction

Same relation on subset of coefficients: REDUCE THE DIM

Attack details (I)

- Signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ invertible (probability $(1 - 1/q)^n \approx 96\%$):

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod{q}$$

WLOG, $b = 0$ (equivalent keys)

- $\mathbf{s}_1$ is short $\implies$ $\mathbf{v}$ very close to lattice

$$L = \mathrm{Span}(q\mathbb{Z}^n, (\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i)_{0 \leqslant i \leqslant m-1})$$

- $\dim(L) = n$ too large to apply lattice reduction

$\boxed{\text{Same relation on subset of coefficients: REDUCE THE DIM}}$

Attack details (I)

- Signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ invertible (probability $(1-1/q)^n \approx 96\%$):

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod q$$

WLOG, $b = 0$ (equivalent keys)

- $\mathbf{s}_1$ is short $\implies$ $\mathbf{v}$ very close to lattice

$$L = \mathrm{Span}(q\mathbb{Z}^n, (\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i)_{0 \leqslant i \leqslant m-1})$$

- $\dim(L) = n$ too large to apply lattice reduction

Same relation on subset of coefficients: REDUCE THE DIM

Attack details (I)

- Signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ invertible (probability $(1-1/q)^n \approx 96\%$):

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod q$$

WLOG, $b = 0$ (equivalent keys)

- $\mathbf{s}_1$ is short $\implies$ $\mathbf{v}$ very close to lattice

$$L = \mathrm{Span}(q\mathbb{Z}^n, (\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i)_{0 \leqslant i \leqslant m-1})$$

- $\dim(L) = n$ too large to apply lattice reduction

Same relation on subset of coefficients: REDUCE THE DIM

Attack details (I)

- Signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ invertible (probability $(1 - 1/q)^n \approx 96\%$):

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod{q}$$

WLOG, $b = 0$ (equivalent keys)

- $\mathbf{s}_1$ is short $\implies$ $\mathbf{v}$ very close to lattice

$$L = \mathrm{Span}(q\mathbb{Z}^n, (\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i)_{0 \leqslant i \leqslant m-1})$$

- $\dim(L) = n$ too large to apply lattice reduction

$\boxed{\text{Same relation on subset of coefficients: } \textsc{Reduce the dim}}$

Attack details (II)

- Subset $I \subset \{0, \dots, n-1\}$ of cardinal ℓ $\varphi_I \colon \mathbb{Z}^n \to \mathbb{Z}^I$ projection

- $\varphi_I(\mathbf{v})$ close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$
  If ℓ large enough, difference should be $\varphi_I(\mathbf{s}_1)$.

- CVP using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log\sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log\sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, $\ell \approx 1.09 \cdot m$

Attack details (II)

- Subset $I \subset \{0, \ldots, n-1\}$ of cardinal ℓ $\varphi_I \colon \mathbb{Z}^n \to \mathbb{Z}^I$ projection

- $\varphi_I(\mathbf{v})$ close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$
  If ℓ large enough, difference should be $\varphi_I(\mathbf{s}_1)$.

- CVP using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log \sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log \sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, $\ell \approx 1.09 \cdot m$

Attack details (II)

- Subset $I \subset \{0, \ldots, n-1\}$ of cardinal ℓ $\varphi_I \colon \mathbb{Z}^n \to \mathbb{Z}^I$ projection

- $\varphi_I(\mathbf{v})$ close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$
  If ℓ large enough, difference should be $\varphi_I(\mathbf{s}_1)$.

- CVP using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log \sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log \sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, $\ell \approx 1.09 \cdot m$

Attack details (II)

- Subset $I \subset \{0, \dots, n-1\}$ of cardinal ℓ $\varphi_I \colon \mathbb{Z}^n \to \mathbb{Z}^I$ projection

- $\varphi_I(\mathbf{v})$ close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$
  If ℓ large enough, difference should be $\varphi_I(\mathbf{s}_1)$.

- CVP using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log \sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log \sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, $\ell \approx 1.09 \cdot m$

Attack details (III)

- In practice: Works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Apply the attack for several choices of l to recover all of s_1, and subsequently s_2: full key recovery with one faulty signature!

Fault iteration $m =$	2	5	10	20	40	60	80	100
Theoretical min dim $\ell_{\min}$	3	6	11	22	44	66	88	110
Dim ℓ (experimental)	3	6	12	24	50	80	110	140
Reduction algorithm	LLL	LLL	LLL	LLL	BKZ–20	BKZ–25	BKZ–25	BKZ–25
Success proba. (%)	100	99	100	100	100	100	100	98
Time recovery ℓ coeffs. (s)	0.002	0.005	0.022	0.23	7.3	119	941	33655
Time full key recovery	0.5 s	0.5 s	1 s	5 s	80 s	14 min	80 min	38 h

Attack details (III)

- In practice: Works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$

- Apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one faulty signature!

Fault iteration $m =$	2	5	10	20	40	60	80	100
Theoretical min dim ℓ_{min}	3	6	11	22	44	66	88	110
Dim ℓ (experimental)	3	6	12	24	50	80	110	140
Reduction algorithm	LLL	LLL	LLL	LLL	BKZ–20	BKZ–25	BKZ–25	BKZ–25
Success proba. (%)	100	99	100	100	100	100	100	98
Time recovery ℓ coeffs. (s)	0.002	0.005	0.022	0.23	7.3	119	941	33655
Time full key recovery	0.5 s	0.5 s	1 s	5 s	80 s	14 min	80 min	38 h

Attack in a nutshell

- **Step 1**: Fault on the generation of the fresh element y_1.
- **Step 2**: Find parts of the secret with multiple CVP instances.
- **Step 3**: Recombine them to do a full key recovery.

Fault iteration $m =$	2	5	10	20	40	60	80	100
Time full key recovery	0.5 s	0.5 s	1 s	5 s	80 s	14 min	80 min	38 h

GPV-Based scheme

- Variant of Ducas-Lyubashevsky-Prest based on GPV-style lattice trapdoors.
- Defined once again over $\mathcal{R} = \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$
- Secret key:

$$\mathbf{B} \leftarrow \begin{pmatrix} \mathbf{M_g} & -\mathbf{M_f} \\ \mathbf{M_G} & -\mathbf{M_F} \end{pmatrix} \in \mathbb{Z}^{2n \times 2n}$$

for $\mathbf{f} \leftarrow D^n_{\sigma_0}$, $\mathbf{g} \leftarrow D^n_{\sigma_0}$

$$f \cdot G - g \cdot F = q$$

Sign and Verify

function $\textsc{Sign}(\mu, sk = \mathbf{B})$
    $\mathbf{c} \leftarrow H(\mu) \in \mathbb{Z}_q^n$
    $(\mathbf{y}, \mathbf{z}) \leftarrow (\mathbf{c}, \mathbf{0}) - \textsc{GaussianSampler}(\mathbf{B}, \sigma, (\mathbf{c}, \mathbf{0}))$ $\triangleright$ $\mathbf{y}, \mathbf{z}$ *are short and satisfy* $\mathbf{y} + \mathbf{z} \cdot \mathbf{h} = \mathbf{c} \bmod q$
    return $\mathbf{z}$
end function

function $\textsc{Verify}(\mu, pk = \mathbf{h}, \mathbf{z})$
    accept iff $\|\mathbf{z}\|_2 + \|H(\mu) - \mathbf{z} \cdot \mathbf{h}\|_2 \leqslant \sigma\sqrt{2n}$
end function

Sign and Verify

function $\textsc{Sign}(\mu, sk = \mathbf{B})$

  $\mathbf{c} \leftarrow H(\mu) \in \mathbb{Z}_q^n$

  $(\mathbf{y}, \mathbf{z}) \leftarrow (\mathbf{c}, \mathbf{0}) - \textsc{GaussianSampler}(\mathbf{B}, \sigma, (\mathbf{c}, \mathbf{0}))$ $\triangleright$ $\mathbf{y}, \mathbf{z}$ *are short and satisfy* $\mathbf{y} + \mathbf{z} \cdot \mathbf{h} = \mathbf{c} \bmod q$

  return $\mathbf{z}$

end function

function $\textsc{Verify}(\mu, pk = \mathbf{h}, \mathbf{z})$

  accept iff $\|\mathbf{z}\|_2 + \|H(\mu) - \mathbf{z} \cdot \mathbf{h}\|_2 \leqslant \sigma\sqrt{2n}$

end function

Gaussian Sampling

function GaussianSampler($\mathbf{B}, \sigma, \mathbf{c}$) $\triangleright$ $\mathbf{b}_i$ (resp. $\widetilde{\mathbf{b}}_i$) are the rows of $\mathbf{B}$ (resp. of its Gram–Schmidt matrix $\widetilde{\mathbf{B}}$)

  $\mathbf{v} \leftarrow \mathbf{0}$

  for $i = 2n$ **down to** 1 **do**

    $c' \leftarrow \langle \mathbf{c}, \widetilde{\mathbf{b}}_{\mathbf{i}} \rangle / \|\widetilde{\mathbf{b}}_i\|_2^2$

    $\sigma' \leftarrow \sigma / \|\widetilde{\mathbf{b}}_i\|_2$

    $r \leftarrow D_{\mathbb{Z},\sigma',c'}$

    $\mathbf{c} \leftarrow \mathbf{c} - r\mathbf{b}_i$ and $\mathbf{v} \leftarrow \mathbf{v} + r\mathbf{b}_i$

  end for

    return $\mathbf{v}$ $\triangleright$ $\mathbf{v}$ sampled according to the lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$

  end function

Gaussian Sampling

function $\textsc{GaussianSampler}(\mathbf{B}, \sigma, \mathbf{c})$ ▷ $\mathbf{b}_i$ (resp. $\widetilde{\mathbf{b}}_i$) are the rows of $\mathbf{B}$ (resp. of its Gram–Schmidt matrix $\widetilde{\mathbf{B}}$)

  $\mathbf{v} \leftarrow \mathbf{0}$

  for $i = 2n$ **down to** 1 **do**

    $c' \leftarrow \langle \mathbf{c}, \widetilde{\mathbf{b}}_{\mathbf{i}} \rangle / \|\widetilde{\mathbf{b}}_i\|_2^2$

    $\sigma' \leftarrow \sigma / \|\widetilde{\mathbf{b}}_i\|_2$

    $r \leftarrow D_{\mathbb{Z}, \sigma', c'}$

    $\mathbf{c} \leftarrow \mathbf{c} - r\mathbf{b}_i$ and $\mathbf{v} \leftarrow \mathbf{v} + r\mathbf{b}_i$

  end for

    return $\mathbf{v}$ ▷ $\mathbf{v}$ sampled according to the lattice Gaussian distribution $D_{\Lambda, \sigma, \mathbf{c}}$

    end function

Attacking the Gaussian sampler

- Correctly generated signature: element of the form

$$\mathbf{z} = \mathbf{R} \cdot \mathbf{f} + \mathbf{r} \cdot \mathbf{F} \in \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$$

- Faults introduced after m iterations of the generation of r, R:

$$\mathbf{z} = r_0 \mathbf{x}^{n-1}\mathbf{F} + r_1 \mathbf{x}^{n-2}\mathbf{F} + \cdots + r_{m-1}\mathbf{x}^{n-m}\mathbf{F}.$$

- Belongs to lattice :

$$L = \mathrm{Span}(\mathbf{x}^{n-i}\mathbf{F})$$

for $1 \leqslant i \leqslant m$.

Attacking the Gaussian sampler

- Correctly generated signature: element of the form

$$\mathbf{z} = \mathbf{R} \cdot \mathbf{f} + \mathbf{r} \cdot \mathbf{F} \in \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$$

- Faults introduced after m iterations of the generation of r, R:

$$\mathbf{z} = r_0 \mathbf{x}^{n-1}\mathbf{F} + r_1 \mathbf{x}^{n-2}\mathbf{F} + \cdots + r_{m-1}\mathbf{x}^{n-m}\mathbf{F}.$$

- Belongs to lattice :

$$L = \mathrm{Span}(\mathbf{x}^{n-i}\mathbf{F})$$

for $1 \leqslant i \leqslant m$.

Attacking the Gaussian sampler

- Correctly generated signature: element of the form

$$\mathbf{z} = \mathbf{R} \cdot \mathbf{f} + \mathbf{r} \cdot \mathbf{F} \in \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$$

- Faults introduced after m iterations of the generation of r, R:

$$\mathbf{z} = r_0 \mathbf{x}^{n-1} \mathbf{F} + r_1 \mathbf{x}^{n-2} \mathbf{F} + \cdots + r_{m-1} \mathbf{x}^{n-m} \mathbf{F}.$$

- Belongs to lattice :

$$L = \mathrm{Span}(\mathbf{x}^{n-i} \mathbf{F})$$

for $1 \leqslant i \leqslant m$.

Multiple faulted signatures?

- $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\ell)}$ faulty signatures.

- With probability $\geqslant \prod_{k=l-m+1}^{+\infty} \frac{1}{\zeta(k)}$ generates L. [*Maze, Rosenthal, Wagner*]

- SVP of L should be one of the $\mathbf{x}^{n-i}\mathbf{F}$ for $1 \leqslant i \leqslant m$.

$\Longrightarrow$ Full recovery of a basis $(\zeta f, \zeta g, \zeta F, \zeta G)$ for a $\zeta = \pm \mathbf{x}^{\alpha}$.
(*equivalent keys*)

Multiple faulted signatures?

- $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(\ell)}$ faulty signatures.
- With probability $\geqslant \prod_{k=l-m+1}^{+\infty} \frac{1}{\zeta(k)}$ generates L. [*Maze, Rosenthal, Wagner*]
- SVP of L should be one of the $\mathbf{x}^{n-i}\mathbf{F}$ for $1 \leqslant i \leqslant m$.

$\implies$ Full recovery of a basis $(\zeta f, \zeta g, \zeta F, \zeta G)$ for a $\zeta = \pm \mathbf{x}^{\alpha}$.
(*equivalent keys*)

Multiple faulted signatures?

- $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\ell)}$ faulty signatures.

- With probability $\geqslant \prod_{k=l-m+1}^{+\infty} \frac{1}{\zeta(k)}$ generates L. [*Maze, Rosenthal, Wagner*]

- SVP of L should be one of the $\mathbf{x}^{n-i}\mathbf{F}$ for $1 \leqslant i \leqslant m$.

$\implies$ Full recovery of a basis $(\zeta f, \zeta g, \zeta F, \zeta G)$ for a $\zeta = \pm \mathbf{x}^{\alpha}$.
(*equivalent keys*)

Multiple faulted signatures?

- $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\ell)}$ faulty signatures.

- With probability $\geqslant \prod_{k=l-m+1}^{+\infty} \frac{1}{\zeta(k)}$ generates L. [*Maze, Rosenthal, Wagner*]

- SVP of L should be one of the $\mathbf{x}^{n-i}\mathbf{F}$ for $1 \leqslant i \leqslant m$.

$\Longrightarrow$ Full recovery of a basis $(\zeta f, \zeta g, \zeta F, \zeta G)$ for a $\zeta = \pm \mathbf{x}^{\alpha}$.
(*equivalent keys*)

In practice

Fault after iteration number $m =$ Lattice reduction algorithm	2 LLL	5 LLL	10 LLL	20 LLL	40 LLL	60 LLL	80 BKZ–20	100 BKZ–20
Success probability for $\ell = m + 1$ (%)	75	77	90	93	94	94	95	95
Avg. CPU time for $\ell = m + 1$ (s)	0.001	0.003	0.016	0.19	2.1	8.1	21.7	104
Success probability for $\ell = m + 2$ (%)	89	95	100	100	99	99	100	100
Avg. CPU time for $\ell = m + 2$ (s)	0.001	0.003	0.017	0.19	2.1	8.2	21.6	146

Conclusion and countermeasures

- Important to investigate implementation attacks on lattice schemes

- Faults against Fiat-Shamir and Hash-And-Sign signatures
  - Among first fault attacks against non-broken lattice signatures
  - Both based on early loop abort
  - One of them recovers the full key with a single faulty sig.
  - Other one: multiple faulty sig., but still on fault per sig.

Conclusion and countermeasures

- Check that the loop ran completely (two loop counters)
- For $\mathbf{y}_1$: check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
- Alternatively: randomize the order of generation of the coefficients (still a bit risky)

Thank you for your attention!