IRISA

LWE without Modular Reduction and Application

Pierre-Alain Fouque

Rennes Univ

IRISA, 2018–5–25

joint work with C. Delaplace, T. Espitau, J. Bootle, M. Tibouchi

Outline

The side-channel leakage of BLISS rejection sampling

LWE over the integers

Outline

The side-channel leakage of BLISS rejection sampling

LWE over the integers

BLISS Rejection Sampling

- The rejection sampling step leaks secret key info through timing side-channels
- More precisely, leakage of two functions of the secret key
  - exact leakage of a quadratic function of the key
  - noisy leakage of a linear function of the key
- In the CCS paper: exploit the quadratic leakage
  - requires relatively few side-channel traces
  - heavy-weight, expensive algebraic number theory
  - can only attack weak keys ($\approx 7\%$)
- Claim: the linear leakage is not useful
  - noisy linear system of dimension $\geq$ original lattice problem
  - so this should not help
- This talk: actually, it is useful!
  - much faster attack than CCS
  - works against all keys
  - drawback: requires more traces

BLISS Rejection Sampling

- The rejection sampling step leaks secret key info through timing side-channels
- More precisely, leakage of two functions of the secret key
  - exact leakage of a quadratic function of the key
  - noisy leakage of a linear function of the key
- In the CCS paper: exploit the quadratic leakage
  - requires relatively few side-channel traces
  - heavy-weight, expensive algebraic number theory
  - can only attack weak keys ($\approx 7\%$)
- Claim: the linear leakage is not useful
  - noisy linear system of dimension $\geq$ original lattice problem
  - so this should not help
- This talk: actually, it is useful!
  - much faster attack than CCS
  - works against all keys
  - drawback: requires more traces

BLISS Rejection Sampling

- The rejection sampling step leaks secret key info through timing side-channels
- More precisely, leakage of two functions of the secret key
  - exact leakage of a quadratic function of the key
  - noisy leakage of a linear function of the key
- In the CCS paper: exploit the quadratic leakage
  - requires relatively few side-channel traces
  - heavy-weight, expensive algebraic number theory
  - can only attack weak keys ($\approx 7\%$)
- Claim: the linear leakage is not useful
  - noisy linear system of dimension $\geq$ original lattice problem
  - so this should not help
- This talk: actually, it is useful!
  - much faster attack than CCS
  - works against all keys
  - drawback: requires more traces

BLISS Rejection Sampling

- The rejection sampling step leaks secret key info through timing side-channels
- More precisely, leakage of two functions of the secret key
  - exact leakage of a quadratic function of the key
  - noisy leakage of a linear function of the key
- In the CCS paper: exploit the quadratic leakage
  - requires relatively few side-channel traces
  - heavy-weight, expensive algebraic number theory
  - can only attack weak keys ($\approx 7\%$)
- Claim: the linear leakage is not useful
  - noisy linear system of dimension $\geq$ original lattice problem
  - so this should not help
- This talk: actually, it is useful!
  - much faster attack than CCS
  - works against all keys
  - drawback: requires more traces

BLISS Rejection Sampling

- The rejection sampling step leaks secret key info through timing side-channels
- More precisely, leakage of two functions of the secret key
  - exact leakage of a quadratic function of the key
  - noisy leakage of a linear function of the key
- In the CCS paper: exploit the quadratic leakage
  - requires relatively few side-channel traces
  - heavy-weight, expensive algebraic number theory
  - can only attack weak keys ($\approx 7\%$)
- Claim: the linear leakage is not useful
  - noisy linear system of dimension $\geq$ original lattice problem
  - so this should not help
- This talk: actually, it is useful!
  - much faster attack than CCS
  - works against all keys
  - drawback: requires more traces

BLISS: the basics

- One of the top contenders for postquantum signatures
- Introduced by Ducas, Durmus, Lepoint and Lyubashevsky at CRYPTO'13
- Implementations on various platforms: desktop computers, microcontrollers/smartcards, FPGAs
- Deployed in the VPN library strongSwan

BLISS: signing and verification keys

- Works in the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(x^n + 1)$, $n = 512$
- Computations modulo the prime $q = 12289$
- Secret key: random sparse $\mathbf{s}_1, \mathbf{s}_2 \in R$ with coefficients in $\{-1, 0, 1\}$
- Verification key: $\mathbf{a} = -\mathbf{s}_2/\mathbf{s}_1 \bmod q$
  - restart if $\mathbf{s}_1$ not invertible

BLISS: signature (simplified)

function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$

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

  $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$ ▷ 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 $1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc}\rangle/\sigma^2\big)$ otherwise **restart**

  $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$

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

end function

BLISS: signature (simplified)

function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$

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

  $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$ ▷ 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 $1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle/\sigma^2\big)$ otherwise **restart**

  $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$

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

end function

BLISS: signature (simplified)

function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$

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

  $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$ ▷ 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 $1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc}\rangle/\sigma^2\big)$ otherwise **restart**

  $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$

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

end function

BLISS: signature (simplified)

```
function Sign(μ, pk = a, sk = S = (s_1, s_2))
    y_1, y_2 ← D^n_{Z,σ}                         ▷ Gaussian sampling
    c ← H(a · y_1 + y_2, μ)                      ▷ special hashing
    choose a random bit b
    z_1 ← y_1 + (−1)^b s_1 c
    z_2 ← y_2 + (−1)^b s_2 c
    continue with probability
  1/(M exp(−‖Sc‖^2/(2σ^2)) cosh(⟨z, Sc⟩/σ^2)) otherwise restart
    z_2^† ← Compress(z_2)
    return (z_1, z_2^†, c)
end function
```

function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$

- $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ — ▷ Gaussian sampling
- $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$ — ▷ 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 $1/\big(M \exp(-\|\mathbf{S}\mathbf{c}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{S}\mathbf{c}\rangle/\sigma^2)\big)$ otherwise **restart**
- $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$
- **return** $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$

end function

BLISS: signature (simplified)

function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$

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

$\quad \mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$ ▷ 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 $1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle / \sigma^2\big)$ otherwise **restart**

$\quad \mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$

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

end function

Overview of the CCS 2017 attack

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient for constrained devices: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ (“relative norm” of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Overview of the CCS 2017 attack

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient for constrained devices: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s}_1}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Overview of the CCS 2017 attack

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient for constrained devices: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Overview of the CCS 2017 attack

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient for constrained devices: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Overview of the CCS 2017 attack

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient for constrained devices: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s}}_1$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Overview of the CCS 2017 attack

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient for constrained devices: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

BLISS rejection sampling

function $\textsc{SampleBernExp}(x)$
  for $i = 0$ to $\ell - 1$ **do**
    if $x_i = 1$ **then**
      Sample $a \leftarrow \mathscr{B}_{c_i}$
      if $a = 0$ **then return** 0
    end if
  end for
  return 1
end function $\triangleright\ x = K - \|\mathbf{Sc}\|^2$

function $\textsc{SampleBernCosh}(x)$
  Sample $a \leftarrow \mathscr{B}_{\exp(-x/f)}$
  if $a = 1$ **then return** 1
  Sample $b \leftarrow \mathscr{B}_{1/2}$
  if $b = 1$ **then restart**
  Sample $c \leftarrow \mathscr{B}_{\exp(-x/f)}$
  if $c = 1$ **then restart**
  return 0
end function $\triangleright\ x = 2 \cdot \langle \mathbf{z}, \mathbf{Sc} \rangle$

Sampling algorithms for the distributions $\mathscr{B}_{\exp(-x/f)}$ and $\mathscr{B}_{1/\cosh(x/f)}$ ($c_i = 2^i/f$ precomputed)

BLISS rejection sampling

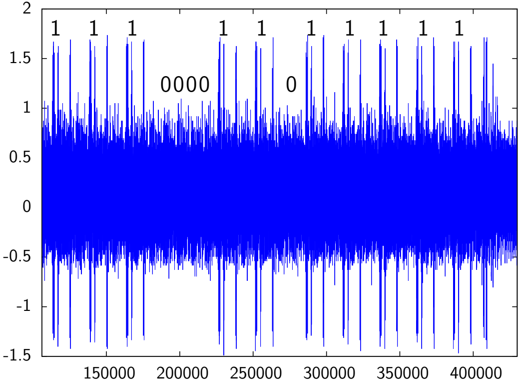
function SampleBernExp(x)
  for $i = 0$ to $\ell - 1$ **do**
    if $x_i = 1$ **then**
      Sample $a \leftarrow \mathscr{B}_{c_i}$
      if $a = 0$ **then return** 0
    end if
  end for
  return 1
end function $\triangleright\ x = K - \|\mathbf{Sc}\|^2$

function SampleBernCosh(x)
  Sample $a \leftarrow \mathscr{B}_{\exp(-x/f)}$
  if $a = 1$ **then return** 1
  Sample $b \leftarrow \mathscr{B}_{1/2}$
  if $b = 1$ **then restart**
  Sample $c \leftarrow \mathscr{B}_{\exp(-x/f)}$
  if $c = 1$ **then restart**
  return 0
end function $\triangleright\ x = 2 \cdot \langle \mathbf{z}, \mathbf{Sc} \rangle$

Sampling algorithms for the distributions $\mathscr{B}_{\exp(-x/f)}$ and $\mathscr{B}_{1/\cosh(x/f)}$ ($c_i = 2^i/f$ precomputed)

Experimental leakage

EMA trace of BLISS rejection sampling on 8-bit AVR for norm $\|\mathbf{Sc}\|^2 = 14404$. One reads the value:
$K - \|\mathbf{Sc}\|^2 = 46539 - 14404 = 32135 = \overline{111110110000111}_2$



What about the inner product leakage? (I)

- Recall the rejection sampling probability of BLISS signing:

$$1 \Bigg/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2} \right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc} \rangle}{\sigma^2} \right) \right),$$

- The exp part of the rejection sampling leaks $\|\mathbf{Sc}\|^2$ and ultimately the relative norm of $\mathbf{s}_1$ and $\mathbf{s}_2$: used in CCS17
- Can't we use the cosh part instead? It directly leaks:

$$\langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c} \rangle + \langle \mathbf{z}_2, \mathbf{s}_2\mathbf{c} \rangle$$

- If we know $(\mathbf{c}, \mathbf{z}_1, \mathbf{z}_2)$, this gives a *linear* relation on the secret: recover everything from around 1024 signatures without breaking a sweat!

What about the inner product leakage? (I)

- Recall the rejection sampling probability of BLISS signing:

$$1 \Bigg/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2} \right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc} \rangle}{\sigma^2} \right) \right),$$

- The exp part of the rejection sampling leaks $\|\mathbf{Sc}\|^2$ and ultimately the relative norm of $\mathbf{s}_1$ and $\mathbf{s}_2$: used in CCS17
- Can't we use the cosh part instead? It directly leaks:

$$\langle \mathbf{z}_1, \mathbf{s}_1 \mathbf{c} \rangle + \langle \mathbf{z}_2, \mathbf{s}_2 \mathbf{c} \rangle$$

- If we know $(\mathbf{c}, \mathbf{z}_1, \mathbf{z}_2)$, this gives a *linear* relation on the secret: recover everything from around 1024 signatures without breaking a sweat!

What about the inner product leakage? (I)

- Recall the rejection sampling probability of BLISS signing:

$$1\Bigg/\left(M\exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle\mathbf{z},\mathbf{Sc}\rangle}{\sigma^2}\right)\right),$$

- The exp part of the rejection sampling leaks $\|\mathbf{Sc}\|^2$ and ultimately the relative norm of $\mathbf{s}_1$ and $\mathbf{s}_2$: used in CCS17
- Can't we use the cosh part instead? It directly leaks:

$$\langle\mathbf{z}_1,\mathbf{s}_1\mathbf{c}\rangle+\langle\mathbf{z}_2,\mathbf{s}_2\mathbf{c}\rangle$$

- If we know $(\mathbf{c},\mathbf{z}_1,\mathbf{z}_2)$, this gives a *linear* relation on the secret: recover everything from around 1024 signatures without breaking a sweat!

What about the inner product leakage? (I)

- Recall the rejection sampling probability of BLISS signing:

$$1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2} \right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc} \rangle}{\sigma^2} \right) \right),$$

- The exp part of the rejection sampling leaks $\|\mathbf{Sc}\|^2$ and ultimately the relative norm of $\mathbf{s}_1$ and $\mathbf{s}_2$: used in CCS17
- Can't we use the cosh part instead? It directly leaks:

$$\langle \mathbf{z}_1, \mathbf{s}_1 \mathbf{c} \rangle + \langle \mathbf{z}_2, \mathbf{s}_2 \mathbf{c} \rangle$$

- If we know $(\mathbf{c}, \mathbf{z}_1, \mathbf{z}_2)$, this gives a *linear* relation on the secret: recover everything from around 1024 signatures without breaking a sweat!

What about the inner product leakage? (II)

- Problem: signatures do not contain $\mathbf{z}_2$, but only a compressed variant $\mathbf{z}_2^\dagger$, and compression is lossy
  - only obtain noisy linear system in the secret key
- First reaction: like LWE in twice the original dimension, so probably hopeless
- Second opinion: not hopeless at all, because there is no modular reduction

What about the inner product leakage? (II)

- Problem: signatures do not contain $\mathbf{z}_2$, but only a compressed variant $\mathbf{z}_2^\dagger$, and compression is lossy
  - only obtain noisy linear system in the secret key
- First reaction: like LWE in twice the original dimension, so probably hopeless
- Second opinion: not hopeless at all, because there is no modular reduction

What about the inner product leakage? (II)

- Problem: signatures do not contain $\mathbf{z}_2$, but only a compressed variant $\mathbf{z}_2^\dagger$, and compression is lossy
  - only obtain noisy linear system in the secret key
- First reaction: like LWE in twice the original dimension, so probably hopeless
- Second opinion: not hopeless at all, because there is no modular reduction

More precise description of the leakage

$$\langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c}\rangle + \langle \mathbf{z}_2, \mathbf{s}_2\mathbf{c}\rangle = \langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c}\rangle + \langle 2^d\mathbf{z}_2^\dagger + (\mathbf{z}_2 - 2^d z_2^\dagger), \mathbf{s}_2\mathbf{c}\rangle$$

$$= \langle \mathbf{z}_1\mathbf{c}^*, \mathbf{s}_1\rangle + \langle 2^d\mathbf{z}_2^\dagger\mathbf{c}^*, \mathbf{s}_2\rangle + \langle \mathbf{z}_2 - 2^d z_2^\dagger, \mathbf{s}_2\mathbf{c}\rangle$$

$$b = \langle \mathbf{a}, \mathbf{s}\rangle + e$$

where

$\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2)$	(secret key)
$\mathbf{a} = \left(\mathbf{z}_1\mathbf{c}^*, 2^d\mathbf{z}_2^\dagger\mathbf{c}^*\right)$	(known from sig.)
$b = \langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c}\rangle + \langle \mathbf{z}_2, \mathbf{s}_2\mathbf{c}\rangle$	(leakage)
$e = \langle \mathbf{z}_2 - 2^d z_2^\dagger, \mathbf{s}_2\mathbf{c}\rangle$	(small unknown value)

More precise description of the leakage

$$\begin{aligned}\langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c}\rangle + \langle \mathbf{z}_2, \mathbf{s}_2\mathbf{c}\rangle &= \langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c}\rangle + \langle 2^d\mathbf{z}_2^\dagger + (\mathbf{z}_2 - 2^d z_2^\dagger), \mathbf{s}_2\mathbf{c}\rangle \\ &= \langle \mathbf{z}_1\mathbf{c}^*, \mathbf{s}_1\rangle + \langle 2^d\mathbf{z}_2^\dagger\mathbf{c}^*, \mathbf{s}_2\rangle + \langle \mathbf{z}_2 - 2^d z_2^\dagger, \mathbf{s}_2\mathbf{c}\rangle \\ b &= \langle \mathbf{a}, \mathbf{s}\rangle + e\end{aligned}$$

where

$$\begin{aligned}\mathbf{s} &= (\mathbf{s}_1, \mathbf{s}_2) && \text{(secret key)} \\ \mathbf{a} &= \left(\mathbf{z}_1\mathbf{c}^*, 2^d\mathbf{z}_2^\dagger\mathbf{c}^*\right) && \text{(known from sig.)} \\ b &= \langle \mathbf{z}_1, \mathbf{s}_1\mathbf{c}\rangle + \langle \mathbf{z}_2, \mathbf{s}_2\mathbf{c}\rangle && \text{(leakage)} \\ e &= \langle \mathbf{z}_2 - 2^d z_2^\dagger, \mathbf{s}_2\mathbf{c}\rangle && \text{(small unknown value)}\end{aligned}$$

Outline

The side-channel leakage of BLISS rejection sampling

LWE over the integers

The Integer LWE problem

- $\mathbf{s}$ secret vector in $\mathbb{Z}^n$
- χ_a, χ_e probability distributions over $\mathbb{Z}$

Integer-LWE Problem

Given m samples $(\mathbf{a}_i, b_i)$ of the form:

$$\mathbf{a}_i \leftarrow \chi_a^n \qquad\qquad b_i = \langle \mathbf{a}, \mathbf{s} \rangle + e \quad (e \leftarrow \chi_e)$$

find $\mathbf{s}$.

Like LWE, without the modular reduction *but* $Var[\chi_e]/Var[\chi_a]$ polynomial in n.

Can we solve this efficiently?

Our main result

Integer-LWE is easy

Suppose χ_a, χ_e are centered distributions of std. dev. σ_a, σ_e. We show that we can recover **s** with m samples for

$$m = O\left(\log n \cdot \left(\frac{\sigma_e}{\sigma_a}\right)^2\right).$$

- In particular, unless σ_e is exponentially larger than σ_a, we can always recover **s** with poly-many samples
- Rigorous results for χ_a, χ_e subgaussian distributions
- Lower bound: $m = \Omega\left(\left(\frac{\sigma_e}{\sigma_a}\right)^2\right)$

Lower Bound on integer-LWE

Let $\mathscr{D}_{\mathbf{s},\chi_a,\chi_e} = \{(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e) : \mathbf{a} \leftarrow \chi_a^n,\ e \leftarrow \chi_e\}$. Given $\mathbf{s} \neq \mathbf{s}' \in \mathbb{Z}^n$, how close are the distributions $\mathscr{D}_{\mathbf{s},\chi_a,\chi_e}$ and $\mathscr{D}_{\mathbf{s}',\chi_a,\chi_e}$?

- We show that when χ_e is either uniform or Gaussian, the statistical distance is bounded by $O(\frac{\sigma_a}{\sigma_e}\|\mathbf{s} - \mathbf{s}'\|)$
- Consequently, we need $\Omega(\frac{1}{\|\mathbf{s}-\mathbf{s}'\|^2}(\frac{\sigma_e}{\sigma_a})^2)$ samples to distinguish those distributions with constant success probability

The least squares approach (I)

- Given $m > n$ integer-LWE samples, we can put them in matrix form:

$$A \in \mathbb{Z}^{m\times n} \qquad\qquad \mathbf{b} = A\mathbf{s} + \mathbf{e} \quad (\mathbf{b}, \mathbf{e} \in \mathbb{Z}^m)$$

- Overdetermined linear system with errors. Least squares: find $\tilde{\mathbf{s}} \in \mathbb{R}^n$ minimizing $\|A\tilde{\mathbf{s}} - \mathbf{b}\|_2^2$
- Solution:

$$\tilde{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{b}$$

- Only makes sense if $A^T A$ invertible, but this should be the case for large m. Indeed: $A^T A = \big(\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big)_{1\le i,j\le n} \in \mathbb{Z}^{n\times n}$
- Law of large numbers: $A^T A \approx E[A^T A]$. Now:

$$E\big[\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big] = \sum_{k=1}^{m} E[a_{ik} a_{jk}] = \begin{cases} m \cdot E[\chi_a]^2 = 0 & i \ne j \\ m \cdot E[\chi_a^2] = m\sigma_a^2 & i = j \end{cases}$$

- Hence, $A^T A \approx m\sigma_a^2 \cdot I_n$ for large m

The least squares approach (I)

- Given $m > n$ integer-LWE samples, we can put them in matrix form:

$$A \in \mathbb{Z}^{m\times n} \qquad \mathbf{b} = A\mathbf{s} + \mathbf{e} \quad (\mathbf{b}, \mathbf{e} \in \mathbb{Z}^m)$$

- Overdetermined linear system with errors. Least squares: find $\tilde{\mathbf{s}} \in \mathbb{R}^n$ minimizing $\|A\tilde{\mathbf{s}} - \mathbf{b}\|_2^2$
- Solution:

$$\tilde{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{b}$$

- Only makes sense if $A^T A$ invertible, but this should be the case for large m. Indeed: $A^T A = \big(\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big)_{1\le i,j\le n} \in \mathbb{Z}^{n\times n}$
- Law of large numbers: $A^T A \approx E[A^T A]$. Now:

$$E\big[\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big] = \sum_{k=1}^{m} E[a_{ik} a_{jk}] = \begin{cases} m \cdot E[\chi_a]^2 = 0 & i \neq j \\ m \cdot E[\chi_a^2] = m\sigma_a^2 & i = j \end{cases}$$

- Hence, $A^T A \approx m\sigma_a^2 \cdot I_n$ for large m

The least squares approach (I)

- Given $m > n$ integer-LWE samples, we can put them in matrix form:

$$A \in \mathbb{Z}^{m\times n} \qquad \mathbf{b} = A\mathbf{s} + \mathbf{e} \quad (\mathbf{b}, \mathbf{e} \in \mathbb{Z}^m)$$

- Overdetermined linear system with errors. Least squares: find $\tilde{\mathbf{s}} \in \mathbb{R}^n$ minimizing $\|A\tilde{\mathbf{s}} - \mathbf{b}\|_2^2$
- Solution:

$$\tilde{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{b}$$

- Only makes sense if $A^T A$ invertible, but this should be the case for large m. Indeed: $A^T A = \big(\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big)_{1\le i,j\le n} \in \mathbb{Z}^{n\times n}$
- Law of large numbers: $A^T A \approx E[A^T A]$. Now:

$$E\big[\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big] = \sum_{k=1}^{m} E[a_{ik}a_{jk}] = \begin{cases} m\cdot E[\chi_a]^2 = 0 & i \neq j \\ m \cdot E[\chi_a^2] = m\sigma_a^2 & i = j \end{cases}$$

- Hence, $A^T A \approx m\sigma_a^2 \cdot I_n$ for large m

The least squares approach (I)

- Given $m > n$ integer-LWE samples, we can put them in matrix form:

$$A \in \mathbb{Z}^{m\times n} \qquad \mathbf{b} = A\mathbf{s} + \mathbf{e} \quad (\mathbf{b}, \mathbf{e} \in \mathbb{Z}^m)$$

- Overdetermined linear system with errors. Least squares: find $\tilde{\mathbf{s}} \in \mathbb{R}^n$ minimizing $\|A\tilde{\mathbf{s}} - \mathbf{b}\|_2^2$
- Solution:

$$\tilde{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{b}$$

- Only makes sense if $A^T A$ invertible, but this should be the case for large m. Indeed: $A^T A = \big(\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big)_{1\le i,j\le n} \in \mathbb{Z}^{n\times n}$
- Law of large numbers: $A^T A \approx E[A^T A]$. Now:

$$E\big[\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big] = \sum_{k=1}^{m} E[a_{ik} a_{jk}] = \begin{cases} m \cdot E[\chi_a]^2 = 0 & i \ne j \\ m \cdot E[\chi_a^2] = m\sigma_a^2 & i = j \end{cases}$$

- Hence, $A^T A \approx m\sigma_a^2 \cdot I_n$ for large m

The least squares approach (I)

- Given $m > n$ integer-LWE samples, we can put them in matrix form:

$$A \in \mathbb{Z}^{m\times n} \qquad \mathbf{b} = A\mathbf{s} + \mathbf{e} \quad (\mathbf{b}, \mathbf{e} \in \mathbb{Z}^m)$$

- Overdetermined linear system with errors. Least squares: find $\tilde{\mathbf{s}} \in \mathbb{R}^n$ minimizing $\|A\tilde{\mathbf{s}} - \mathbf{b}\|_2^2$
- Solution:

$$\tilde{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{b}$$

- Only makes sense if $A^T A$ invertible, but this should be the case for large m. Indeed: $A^T A = \big(\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big)_{1\le i,j\le n} \in \mathbb{Z}^{n\times n}$
- Law of large numbers: $A^T A \approx E[A^T A]$. Now:

$$E\big[\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big] = \sum_{k=1}^{m} E[a_{ik} a_{jk}] = \begin{cases} m \cdot E[\chi_a]^2 = 0 & i \neq j \\ m \cdot E[\chi_a^2] = m\sigma_a^2 & i = j \end{cases}$$

- Hence, $A^T A \approx m\sigma_a^2 \cdot I_n$ for large m

The least squares approach (I)

- Given $m > n$ integer-LWE samples, we can put them in matrix form:

$$A \in \mathbb{Z}^{m\times n} \qquad \mathbf{b} = A\mathbf{s} + \mathbf{e} \quad (\mathbf{b}, \mathbf{e} \in \mathbb{Z}^m)$$

- Overdetermined linear system with errors. Least squares: find $\tilde{\mathbf{s}} \in \mathbb{R}^n$ minimizing $\|A\tilde{\mathbf{s}} - \mathbf{b}\|_2^2$
- Solution:

$$\tilde{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{b}$$

- Only makes sense if $A^T A$ invertible, but this should be the case for large m. Indeed: $A^T A = \big(\langle \mathbf{a}_i, \mathbf{a}_j \rangle\big)_{1\le i,j\le n} \in \mathbb{Z}^{n\times n}$
- Law of large numbers: $A^T A \approx E[A^T A]$. Now:

$$E\big[\langle \mathbf{a}_i, \mathbf{a}_j\rangle\big] = \sum_{k=1}^{m} E[a_{ik}a_{jk}] = \begin{cases} m\cdot E[\chi_a]^2 = 0 & i \ne j \\ m \cdot E[\chi_a^2] = m\sigma_a^2 & i = j \end{cases}$$

- Hence, $A^T A \approx m\sigma_a^2 \cdot I_n$ for large m

The least squares approach (II)

- Claim: $\tilde{\mathbf{s}}$ is an approximation of $\mathbf{s}$
- The difference is a function of A and $\mathbf{e}$:

$$\begin{aligned}\tilde{\mathbf{s}} - \mathbf{s} &= (A^T A)^{-1} A^T \mathbf{b} - \mathbf{s} \\ &= (A^T A)^{-1} A^T \Big(A\mathbf{s} + \mathbf{e} \Big) - \mathbf{s} = (A^T A)^{-1} A^T \mathbf{e}\end{aligned}$$

- Thus, we can bound the Euclidean distance:

$$\begin{aligned}\|\tilde{\mathbf{s}} - \mathbf{s}\|^2 &= \|(A^T A)^{-1} A^T \mathbf{e}\|^2 \\ &\leq \underbrace{\|(A^T A)^{-1/2}\|^2}_{\text{operator norm}} \cdot \|(A^T A)^{-1/2} A^T \mathbf{e}\|^2 \\ &= \lambda_{\min}^{-1} \cdot \mathbf{e}^T A (A^T A)^{-1} A^T \mathbf{e} = \lambda_{\min}^{-1} \cdot \mathbf{e}^T M \mathbf{e}\end{aligned}$$

where $\lambda_{\min} \approx m\sigma_a^2$ smallest eigenvalue of $A^T A$ and $M = A(A^T A)^{-1} A^T$

The least squares approach (II)

- Claim: $\tilde{\mathbf{s}}$ is an approximation of $\mathbf{s}$
- The difference is a function of A and $\mathbf{e}$:

$$\begin{aligned}\tilde{\mathbf{s}} - \mathbf{s} &= (A^T A)^{-1} A^T \mathbf{b} - \mathbf{s} \\ &= (A^T A)^{-1} A^T \left(A\mathbf{s} + \mathbf{e}\right) - \mathbf{s} = (A^T A)^{-1} A^T \mathbf{e}\end{aligned}$$

- Thus, we can bound the Euclidean distance:

$$\begin{aligned}\|\tilde{\mathbf{s}} - \mathbf{s}\|^2 &= \|(A^T A)^{-1} A^T \mathbf{e}\|^2 \\ &\leq \underbrace{\|(A^T A)^{-1/2}\|^2}_{\text{operator norm}} \cdot \|(A^T A)^{-1/2} A^T \mathbf{e}\|^2 \\ &= \lambda_{\min}^{-1} \cdot \mathbf{e}^T A (A^T A)^{-1} A^T \mathbf{e} = \lambda_{\min}^{-1} \cdot \mathbf{e}^T M \mathbf{e}\end{aligned}$$

where $\lambda_{\min} \approx m\sigma_a^2$ smallest eigenvalue of $A^T A$ and $M = A(A^T A)^{-1} A^T$

The least squares approach (II)

- Claim: $\tilde{\mathbf{s}}$ is an approximation of $\mathbf{s}$
- The difference is a function of A and $\mathbf{e}$:

$$\begin{aligned}\tilde{\mathbf{s}} - \mathbf{s} &= (A^T A)^{-1} A^T \mathbf{b} - \mathbf{s} \\ &= (A^T A)^{-1} A^T \Big(A\mathbf{s} + \mathbf{e}\Big) - \mathbf{s} = (A^T A)^{-1} A^T \mathbf{e}\end{aligned}$$

- Thus, we can bound the Euclidean distance:

$$\begin{aligned}\|\tilde{\mathbf{s}} - \mathbf{s}\|^2 &= \|(A^T A)^{-1} A^T \mathbf{e}\|^2 \\ &\leq \underbrace{\|(A^T A)^{-1/2}\|^2}_{\text{operator norm}} \cdot \|(A^T A)^{-1/2} A^T \mathbf{e}\|^2 \\ &= \lambda_{\min}^{-1} \cdot \mathbf{e}^T A (A^T A)^{-1} A^T \mathbf{e} = \lambda_{\min}^{-1} \cdot \mathbf{e}^T M \mathbf{e}\end{aligned}$$

where $\lambda_{\min} \approx m\sigma_a^2$ smallest eigenvalue of $A^T A$ and $M = A(A^T A)^{-1} A^T$

The least squares approach (III)

- Usually, the least square method approximates $\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2$, but we need $\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty^2$ to round each coefficient
- If χ_e is τ_e-subgaussian, $\tilde{\mathbf{s}} - \mathbf{s}$ is $\tau_e/\sqrt{\lambda_{\min}(A^T A)}$-subgaussian
- if $\mathbf{v} \in \mathbb{R}^n$ is τ-subgaussian, $\Pr[\|\mathbf{v}\|_\infty > t] \le 2n \cdot \exp(-t^2/2\tau^2)$
- Then, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] \le 2n \cdot \exp(-\frac{\lambda_{\min}(A^T A)}{8\tau_e^2})$, where $\lambda_{\min}(A^T A) < \frac{m\sigma_a^2}{2}$ whp
- If $m \ge 32\frac{\tau_e^2}{\sigma_a^2}\log(2n)$, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] = 1/2n$

The least squares approach (III)

- Usually, the least square method approximates $\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2$, but we need $\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty^2$ to round each coefficient
- If χ_e is τ_e-subgaussian, $\tilde{\mathbf{s}} - \mathbf{s}$ is $\tau_e/\sqrt{\lambda_{\min}(A^T A)}$-subgaussian
- if $\mathbf{v} \in \mathbb{R}^n$ is τ-subgaussian, $\Pr[\|\mathbf{v}\|_\infty > t] \le 2n \cdot \exp(-t^2/2\tau^2)$
- Then, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] \le 2n \cdot \exp(-\frac{\lambda_{\min}(A^T A)}{8\tau_e^2})$, where $\lambda_{\min}(A^T A) < \frac{m\sigma_a^2}{2}$ whp
- If $m \ge 32\frac{\tau_e^2}{\sigma_a^2}\log(2n)$, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] = 1/2n$

The least squares approach (III)

- Usually, the least square method approximates $\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2$, but we need $\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty^2$ to round each coefficient
- If χ_e is τ_e-subgaussian, $\tilde{\mathbf{s}} - \mathbf{s}$ is $\tau_e/\sqrt{\lambda_{\min}(A^T A)}$-subgaussian
- if $\mathbf{v} \in \mathbb{R}^n$ is τ-subgaussian, $\Pr[\|\mathbf{v}\|_\infty > t] \le 2n \cdot \exp(-t^2/2\tau^2)$
- Then, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] \le 2n \cdot \exp(-\frac{\lambda_{\min}(A^T A)}{8\tau_e^2})$, where $\lambda_{\min}(A^T A) < \frac{m\sigma_a^2}{2}$ whp
- If $m \ge 32\frac{\tau_e^2}{\sigma_a^2}\log(2n)$, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] = 1/2n$

The least squares approach (III)

- Usually, the least square method approximates $\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2$, but we need $\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty^2$ to round each coefficient
- If χ_e is τ_e-subgaussian, $\tilde{\mathbf{s}} - \mathbf{s}$ is $\tau_e/\sqrt{\lambda_{\min}(A^T A)}$-subgaussian
- if $\mathbf{v} \in \mathbb{R}^n$ is τ-subgaussian, $\Pr[\|\mathbf{v}\|_\infty > t] \leq 2n \cdot \exp(-t^2/2\tau^2)$
- Then, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] \leq 2n \cdot \exp(-\frac{\lambda_{\min}(A^T A)}{8\tau_e^2})$, where $\lambda_{\min}(A^T A) < \frac{m\sigma_a^2}{2}$ whp
- If $m \geq 32\frac{\tau_e^2}{\sigma_a^2}\log(2n)$, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] = 1/2n$

The least squares approach (III)

- Usually, the least square method approximates $\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2$, but we need $\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty^2$ to round each coefficient
- If χ_e is τ_e-subgaussian, $\tilde{\mathbf{s}} - \mathbf{s}$ is $\tau_e/\sqrt{\lambda_{\min}(A^T A)}$-subgaussian
- if $\mathbf{v} \in \mathbb{R}^n$ is τ-subgaussian, $\Pr[\|\mathbf{v}\|_\infty > t] \leq 2n \cdot \exp(-t^2/2\tau^2)$
- Then, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] \leq 2n \cdot \exp(-\frac{\lambda_{\min}(A^T A)}{8\tau_e^2})$, where $\lambda_{\min}(A^T A) < \frac{m\sigma_a^2}{2}$ whp
- If $m \geq 32\frac{\tau_e^2}{\sigma_a^2}\log(2n)$, $\Pr[\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty > 1/2] = 1/2n$

Conclusion

- Linear Regression + Rounding can be seen as equivalent to Babai algorithm
- Nearest Plane Algorithm is not always better in practice when the lattice is nearly orthogonal
- Taking into account sparsity of the BLISS secret key is not easy even with linear programming in practice (similar to compressed sensing)

Conclusion

- Linear Regression + Rounding can be seen as equivalent to Babai algorithm
- Nearest Plane Algorithm is not always better in practice when the lattice is nearly orthogonal
- Taking into account sparsity of the BLISS secret key is not easy even with linear programming in practice (similar to compressed sensing)

Conclusion

- Linear Regression + Rounding can be seen as equivalent to Babai algorithm
- Nearest Plane Algorithm is not always better in practice when the lattice is nearly orthogonal
- Taking into account sparsity of the BLISS secret key is not easy even with linear programming in practice (similar to compressed sensing)