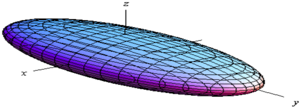
On error distributions in ring-based LWE



Wouter Castryck[1,2], Ilia Iliashenko[1], Frederik Vercauteren[1,3]

[1] COSIC, KU Leuven
[2] Ghent University
[3] Open Security Research

PQCRYPTO
ICT-645622

HEAT

Motivation for LWE

1981 A basic concept of a quantum computer by Feynman

1994 Shor's algorithm

- Factorization and DLP are easy
- Broken: RSA, Diffie-Hellman, ECDLP etc.

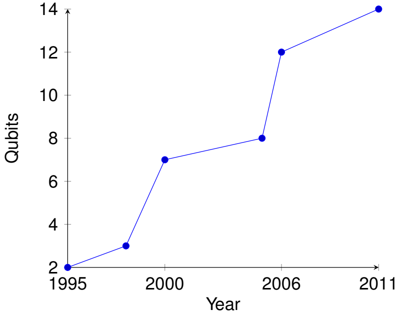
Motivation for LWE

1981 A basic concept of a quantum computer by Feynman

1994 Shor's algorithm

- Factorization and DLP are easy
- Broken: RSA, Diffie-Hellman, ECDLP etc.

1995 First quantum logic gate by Monroe, Meekhof, King, Itano and Wineland



Motivation for LWE

2016 CNSA Suite and Quantum Computing FAQ by NSA

> "Many experts predict a quantum computer capable of effectively breaking public key cryptography within a few decades, and therefore NSA believes it is important to address that concern."

NIST report on post-quantum crypto

> "We must begin now to prepare our information security systems to be able to resist quantum computing."

Learning With Errors (LWE)

The LWE problem (Regev, ‘05): solve a linear system with noise

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n} \\ a_{21} & a_{22} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix}$$

over a finite field $\mathbb{F}_q$ for a secret $(s_1, s_2, \dots, s_n) \in \mathbb{F}_q^n$ where

- a modulus $q = \text{poly}(n)$
- the $a_{ij} \in \mathbb{F}_q$ are chosen uniformly randomly,
- an adversary can ask for new equations ($m > n$).

Learning With Errors (LWE)

The LWE problem is easy when $\forall e_i = 0$.

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n} \\ a_{21} & a_{22} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$$

Gaussian elimination solves the problem.

Learning With Errors (LWE)

The LWE problem is easy when $\forall e_i = 0$.

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n} \\ a_{21} & a_{22} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$$

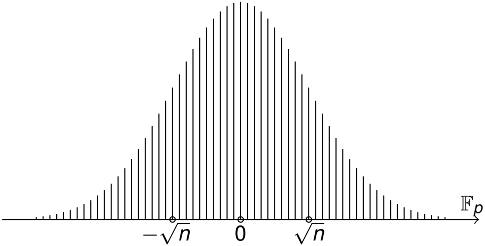
Gaussian elimination solves the problem.
Otherwise, LWE might be hard.

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n} \\ a_{21} & a_{22} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix}$$

Gaussian elimination amplifies errors.

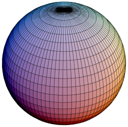
Learning With Errors (LWE)

The errors e_i are sampled independently from a Gaussian with standard deviation $\sigma \gtrsim \sqrt{n}$:



When viewed jointly, the error vector

$$\begin{pmatrix} e_1 \\ \vdots \\ e_m \end{pmatrix}$$

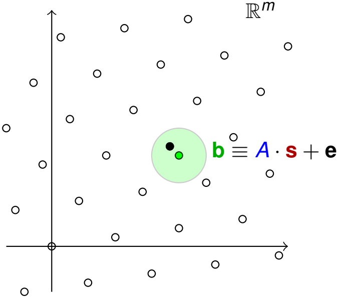


is sampled from a spherical Gaussian.

Learning With Errors (LWE)

LWE is tightly related to classical lattice problems.

- Bounding Distance Decoding (BDD)



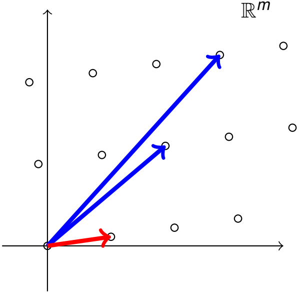
Given $\mathbf{b}$, find a closest point of the q-ary lattice

$$\{\mathbf{w} \in \mathbb{Z}^m \,|\, \exists \mathbf{s} \in \mathbb{Z}^n : \mathbf{w} \equiv A \cdot \mathbf{s} \bmod q\}$$

Learning With Errors (LWE)

LWE is tightly related to classical lattice problems.

- Shortest Vector Problem (SVP)

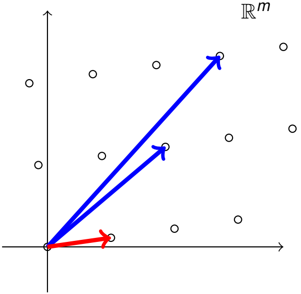


Given a basis, find a shortest non-zero vector of the lattice.

Learning With Errors (LWE)

LWE is tightly related to classical lattice problems.

- Shortest Vector Problem (SVP)



Given a basis, find a shortest non-zero vector of the lattice.

- LWE is at least as hard as worst-case SVP-type problems (Regev‘05, Peikert‘09).
- Not known to be broken by quantum computers.

Learning With Errors (LWE)

Known attacks for $q = \mathrm{poly}(n)$:

	Time	Samples
Trial and error	$2^{O(n \log n)}$	$O(n)$
Blum, Kalai, Wasserman ‘03	$2^{O(n)}$	$2^{O(n)}$
Arora, Ge ‘11	$2^{O(\sigma^2 \log n)}$	$2^{O(\sigma^2 \log n)}$

Learning With Errors (LWE)

Known attacks for $q = \text{poly}(n)$:

	Time	Samples
Trial and error	$2^{O(n \log n)}$	$O(n)$
Blum, Kalai, Wasserman ‘03	$2^{O(n)}$	$2^{O(n)}$
Arora, Ge ‘11	$2^{O(\sigma^2 \log n)}$	$2^{O(\sigma^2 \log n)}$

Idea: if all errors (almost) certainly lie in $\{-T, \ldots, T\}$, then

$$\prod_{i=-T}^{T} (a_1 s_1 + a_2 s_2 + \cdots + a_n s_n - b + i) = 0.$$

View as linear system of equations in $\approx n^{2T}$ monomials.

Learning With Errors (LWE)

Known attacks for $q = \mathrm{poly}(n)$:

	Time	Samples
Trial and error	$2^{O(n \log n)}$	$O(n)$
Blum, Kalai, Wasserman '03	$2^{O(n)}$	$2^{O(n)}$
Arora, Ge '11	$2^{O(\sigma^2 \log n)}$	$\underline{2^{O(\sigma^2 \log n)}}$

Idea: if all errors (almost) certainly lie in $\{-T, \dots, T\}$, then

$$\prod_{i=-T}^{T} (a_1 s_1 + a_2 s_2 + \cdots + a_n s_n - b + i) = 0.$$

View as linear system of equations in $\approx n^{2T}$ monomials.

Learning With Errors (LWE)

Application: public-key encryption of a bit (Regev'05).

- Private key: $\mathbf{s} \in \mathbb{F}_q^n$.
- Public key pair: $(A, \mathbf{b} = A \cdot \mathbf{s} + \mathbf{e})$.

Learning With Errors (LWE)

Application: public-key encryption of a bit (Regev'05).

- Private key: $\mathbf{s} \in \mathbb{F}_q^n$.
- Public key pair: $(A, \mathbf{b} = A \cdot \mathbf{s} + \mathbf{e})$.
- Encrypt: pick random row vector $\mathbf{r}^T \in \{0,1\}^m \subset \mathbb{F}_q^m$. Output the pair

$$\mathbf{c}^T := \mathbf{r}^T \cdot A \quad \text{and} \quad \mathbf{d} := \begin{cases} \mathbf{r}^T \cdot \mathbf{b} & \text{if the bit is 0,} \\ \mathbf{r}^T \cdot \mathbf{b} + \lfloor q/2 \rfloor & \text{if the bit is 1.} \end{cases}$$

Learning With Errors (LWE)

Application: public-key encryption of a bit (Regev'05).

- Private key: $\mathbf{s} \in \mathbb{F}_q^n$.
- Public key pair: $(A, \mathbf{b} = A \cdot \mathbf{s} + \mathbf{e})$.
- Encrypt: pick random row vector $\mathbf{r}^T \in \{0,1\}^m \subset \mathbb{F}_q^m$.
  Output the pair

$$\mathbf{c}^T := \mathbf{r}^T \cdot A \quad \text{and} \quad \mathbf{d} := \begin{cases} \mathbf{r}^T \cdot \mathbf{b} & \text{if the bit is 0,} \\ \mathbf{r}^T \cdot \mathbf{b} + \lfloor q/2 \rfloor & \text{if the bit is 1.} \end{cases}$$

- Decryption of pair $\mathbf{c}^T, \mathbf{d}$: compute

$$\mathbf{d} - \mathbf{c}^T \cdot \mathbf{s} = \mathbf{d} - \mathbf{r}^T \cdot A \cdot \mathbf{s} = \mathbf{d} - \mathbf{r}^T \mathbf{b} - \underset{\substack{\uparrow \\ \text{small enough}}}{\mathbf{r}^T \mathbf{e}} \approx \begin{cases} 0 & \text{if bit was 0,} \\ \lfloor q/2 \rfloor & \text{if bit was 1.} \end{cases}$$

Learning With Errors (LWE)

- Features:
  - Hardness reduction from classical lattice problems
  - Linear operations
    - simple and efficient implementation
    - highly parallelizable
  - Source of exciting applications
    - FHE, attribute-based encryption for arbitrary access policies, general-purpose code obfuscation

Learning With Errors (LWE)

- Features:
  - Hardness reduction from classical lattice problems
  - Linear operations
    - simple and efficient implementation
    - highly parallelizable
  - Source of exciting applications
    - FHE, attribute-based encryption for arbitrary access policies, general-purpose code obfuscation
- Drawback: key size.
  - To hide the secret one needs an entire linear system:

$$\underset{\substack{\uparrow\\ m\log p}}{\begin{pmatrix} b_1\\ b_2\\ \vdots\\ b_m \end{pmatrix}} = \underset{\substack{\uparrow\\ mn\log p}}{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n}\\ a_{21} & a_{22} & \dots & a_{2,n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{m1} & a_{m2} & \dots & a_{m,n} \end{pmatrix}} \cdot \underset{\substack{\uparrow\\ n\log p}}{\begin{pmatrix} s_1\\ s_2\\ \vdots\\ s_n \end{pmatrix}} + \begin{pmatrix} e_1\\ e_2\\ \vdots\\ e_n \end{pmatrix}$$

Ring-based LWE

- Identify vector space

$$\mathbb{F}_q^n \qquad \text{with} \qquad \mathcal{R}_q = \mathbb{Z}[x]/(q, f(x))$$

for some irreducible monic $f(x) \in \mathbb{Z}[x]$ s.t. $\deg f = n$, by viewing

$$(s_1, s_2, \ldots, s_n) \quad \text{as} \quad s_1 + s_2 x + \cdots + s_n x^{n-1}.$$

Ring-based LWE

- Identify vector space

$$\mathbb{F}_q^n \qquad \text{with} \qquad \mathcal{R}_q = \mathbb{Z}[x]/(q, f(x))$$

for some irreducible monic $f(x) \in \mathbb{Z}[x]$ s.t. $\deg f = n$, by viewing

$$(s_1, s_2, \ldots, s_n) \quad \text{as} \quad s_1 + s_2 x + \cdots + s_n x^{n-1}.$$

- Use samples of the form

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

with $A_{\mathbf{a}}$ the matrix of multiplication by some random $\mathbf{a}(x) = a_1 + a_2 x + \cdots + a_n x^{n-1}$.

Ring-based LWE

- Identify vector space

$$\mathbb{F}_q^n \qquad \text{with} \qquad \mathcal{R}_q = \mathbb{Z}[x]/(q, f(x))$$

  for some irreducible monic $f(x) \in \mathbb{Z}[x]$ s.t. $\deg f = n$, by viewing

$$(s_1, s_2, \ldots, s_n) \quad \text{as} \quad s_1 + s_2 x + \cdots + s_n x^{n-1}.$$

- Use samples of the form

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

  with $A_{\mathbf{a}}$ the matrix of multiplication by some random $\mathbf{a}(x) = a_1 + a_2 x + \cdots + a_n x^{n-1}$.

- Store $\mathbf{a}(x)$ rather than $A_{\mathbf{a}}$: saves factor n.

Ring-based LWE

Example:

- if $f(x) = x^n + 1$, then $A_{\mathbf{a}}$ is the anti-circulant matrix

$$\begin{pmatrix} a_1 & -a_n & \dots & -a_3 & -a_2 \\ a_2 & a_1 & \dots & -a_4 & -a_3 \\ a_3 & a_2 & \dots & -a_5 & -a_4 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_n & a_{n-1} & \dots & a_2 & a_1 \end{pmatrix}$$

of which it suffices to store the first column.

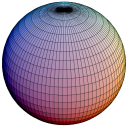
Ring-based LWE

Direct ring-based analogue of LWE-sample would read

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

with the e_i sampled independently from

$$\mathcal{N}(0, \sigma)$$



for some fixed small $\sigma = \sigma(n)$.

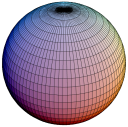
Ring-based LWE

Direct ring-based analogue of LWE-sample would read

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

with the e_i sampled independently from

$$\mathcal{N}(0, \sigma)$$



for some fixed small $\sigma = \sigma(n)$.

This is not Ring-LWE!

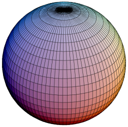
Ring-based LWE

Direct ring-based analogue of LWE-sample would read

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

with the e_i sampled independently from

$$\mathcal{N}(0, \sigma)$$



for some fixed small $\sigma = \sigma(n)$.

This is not Ring-LWE!

- Not backed up by hardness statement.

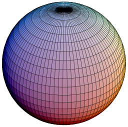
Ring-based LWE

Direct ring-based analogue of LWE-sample would read

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

with the e_i sampled independently from

$$\mathcal{N}(0, \sigma)$$



for some fixed small $\sigma = \sigma(n)$.

This is not Ring-LWE!

- Not backed up by hardness statement.
- Sometimes called Poly-LWE.

Ring-LWE

So what is Ring-LWE according to [LPR10]? Samples look like

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \qquad\qquad \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

Ring-LWE

So what is Ring-LWE according to [LPR10]? Samples look like

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + A_{f'(x)} \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

where

- B is the canonical embedding matrix.
- $A_{f'(x)}$ compensates for the fact that one actually picks secrets from the dual.

Ring-LWE

So what is Ring-LWE according to [LPR10]? Samples look like

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + A_{f'(x)} \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

where

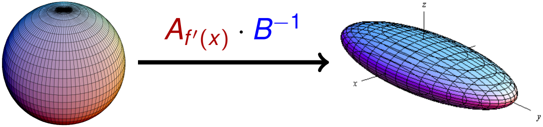
- B is the canonical embedding matrix.
- $A_{f'(x)}$ compensates for the fact that one actually picks secrets from the dual.

Hardness reduction from ideal lattice problems.

Ring-LWE

Note:

- factor $A_{f'(x)} \cdot B^{-1}$ might skew the error distribution,

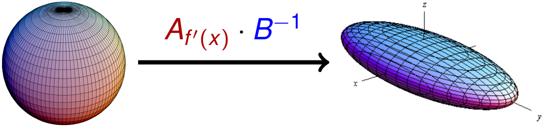


$A_{f'(x)} \cdot B^{-1}$

Ring-LWE

Note:

- factor $A_{f'(x)} \cdot B^{-1}$ might skew the error distribution,



$A_{f'(x)} \cdot B^{-1}$

- but also scales it!
  - det $A_{f'(x)} = \Delta$ with

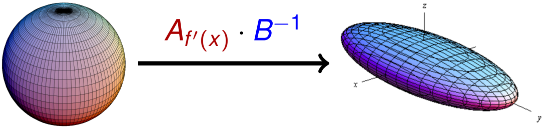
$$\Delta = \left|\operatorname{disc} f(x)\right|, \quad \leftarrow \text{could be huge}$$

  - det $B^{-1} = 1/\sqrt{\Delta}$.

Ring-LWE

Note:

- factor $A_{f'(x)} \cdot B^{-1}$ might skew the error distribution,



- but also scales it!
  - $\det A_{f'(x)} = \Delta$ with

$$\Delta = \left|\operatorname{disc} f(x)\right|, \quad \leftarrow \text{could be huge}$$

  - $\det B^{-1} = 1/\sqrt{\Delta}$.

So “on average”, each e_i is scaled up by $\sqrt{\Delta}^{1/n}$ . . .

- . . . but remember: skewness.

Scaled Canonical Gaussian ring-based LWE

$A_{f'(x)}$ is changed to a scalar λ

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \lambda \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}.$$

The natural choice is $\lambda = |\Delta|^{1/n}$.

- So $\det A_\lambda = |\Delta|$.

Scaled Canonical Gaussian ring-based LWE

$A_{f'(x)}$ is changed to a scalar λ

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \lambda \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}.$$

The natural choice is $\lambda = |\Delta|^{1/n}$.

- So $\det A_\lambda = |\Delta|$.

SCG-LWE = Ring-LWE for 2^m-cyclotomic fields:

- $f'(x) = 2^{m-1} x^{2^{m-1}-1}$,
- $\lambda = 2^{m-1}$,
- So $A_{f'(x)} = M \cdot A_\lambda$ for some matrix $M \in GL_n(\mathbb{Z})$.

Main result

For SCG ring-based LWE with parameters:

- $n = 2^\ell$ for some $\ell \in \mathbb{N}$,
- a modulus $q = \mathrm{poly}(n)$,
- an error distribution with $\sigma = \mathrm{poly}(n)$,
- an underlying field $K = \mathbb{Q}(\sqrt{p_1}, \sqrt{p_2}, \dots, \sqrt{p_\ell})$,
  - a square-free $m = \prod p_i \geq (2\sigma\sqrt{n \log n})^{2/\varepsilon}$ for some $\varepsilon > 0$,
  - $\forall i : p_i \equiv 1 \mod 4$, so $\Delta_K = m^{n/2}$,
- a scaling parameter $\lambda' = \lambda / |\Delta_K|^{\varepsilon/n}$

Main result

For SCG ring-based LWE with parameters:

- $n = 2^\ell$ for some $\ell \in \mathbb{N}$,
- a modulus $q = \mathrm{poly}(n)$,
- an error distribution with $\sigma = \mathrm{poly}(n)$,
- an underlying field $K = \mathbb{Q}(\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_\ell})$,
  - a square-free $m = \prod p_i \geq (2\sigma\sqrt{n \log n})^{2/\varepsilon}$ for some $\varepsilon > 0$,
  - $\forall i : p_i \equiv 1 \mod 4$, so $\Delta_K = m^{n/2}$,
- a scaling parameter $\lambda' = \lambda / |\Delta_K|^{\varepsilon/n}$

there exist an attack with

Time: $\mathrm{poly}(n \cdot \log(q))$ **Space:** $O(n)$ samples

Main result

For SCG ring-based LWE with parameters:

- $n = 2^\ell$ for some $\ell \in \mathbb{N}$,
- a modulus $q = \mathrm{poly}(n)$,
- an error distribution with $\sigma = \mathrm{poly}(n)$,
- an underlying field $K = \mathbb{Q}(\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_\ell})$,
    - a square-free $m = \prod p_i \geq (2\sigma\sqrt{n \log n})^{2/\varepsilon}$ for some $\varepsilon > 0$,
    - $\forall i : p_i \equiv 1 \mod 4$, so $\Delta_K = m^{n/2}$,
- a scaling parameter $\lambda' = \lambda / |\Delta_K|^{\varepsilon/n}$

there exist an attack with

Time: $\mathrm{poly}(n \cdot \log(q))$ **Space:** $O(n)$ samples

$\lambda' = \lambda / |\Delta_K|^{1/2n}$ appears in ELOS‘15, CLS‘15, CLS‘16.

Main result

Tensor structure:

- $K = K_1 \otimes_{\mathbb{Q}} K_2 \otimes_{\mathbb{Q}} \cdots \otimes_{\mathbb{Q}} K_\ell$,
    - where $K_i = \mathbb{Q}(\sqrt{p_i})$
- The ring of integers $R = R_1 \otimes_{\mathbb{Z}} R_2 \otimes_{\mathbb{Z}} \cdots \otimes_{\mathbb{Z}} R_\ell$,
    - where $R_i = \mathbb{Z}[(1 + \sqrt{p_i})/2]$
- The dual $R^\vee = \frac{1}{\sqrt{m}} R = R_1^\vee \otimes_{\mathbb{Z}} R_2^\vee \otimes_{\mathbb{Z}} \cdots \otimes_{\mathbb{Z}} R_\ell^\vee$

Main result

Tensor structure:

- $K = K_1 \otimes_{\mathbb{Q}} K_2 \otimes_{\mathbb{Q}} \cdots \otimes_{\mathbb{Q}} K_\ell$,
  - where $K_i = \mathbb{Q}(\sqrt{p_i})$
- The ring of integers $R = R_1 \otimes_{\mathbb{Z}} R_2 \otimes_{\mathbb{Z}} \cdots \otimes_{\mathbb{Z}} R_\ell$,
  - where $R_i = \mathbb{Z}[(1+\sqrt{p_i})/2]$
- The dual $R^\vee = \frac{1}{\sqrt{m}} R = R_1^\vee \otimes_{\mathbb{Z}} R_2^\vee \otimes_{\mathbb{Z}} \cdots \otimes_{\mathbb{Z}} R_\ell^\vee$

So $\lambda \cdot B^{-1}$ is a Kronecker product of corresponding matrices in underlying quadratic fields K_i

$$\begin{pmatrix} \frac{-1+\sqrt{p_i}}{2} & \frac{1+\sqrt{p_i}}{2} \\ 1 & -1 \end{pmatrix}$$

Main result

Note

$$\begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{-1+\sqrt{p_i}}{2} & \frac{1+\sqrt{p_i}}{2} \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \end{pmatrix}$$

and through the Kronecker product

$$\begin{pmatrix} 0 & 0 & \dots & 1 \end{pmatrix} \cdot \lambda \cdot B^{-1} = \mathbf{d} \in \{1, -1\}^n$$

Main result

Note

$$\begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{-1+\sqrt{p_i}}{2} & \frac{1+\sqrt{p_i}}{2} \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \end{pmatrix}$$

and through the Kronecker product

$$\begin{pmatrix} 0 & 0 & \dots & 1 \end{pmatrix} \cdot \lambda \cdot B^{-1} = \mathbf{d} \in \{1, -1\}^n$$

Applying to an error term of

$$\mathbf{b} = A_{\mathbf{a}} \cdot \mathbf{s} + \lambda' \cdot B^{-1} \cdot \mathbf{e}$$

we have

$$|\Delta_K|^{-\varepsilon/n} \cdot \mathbf{d} \cdot \begin{pmatrix} e_1 & e_2 & \dots & e_n \end{pmatrix}^T = \omega.$$

Main result

ω is distributed by Gaussian with the standard deviation

$$\frac{\sqrt{n}\cdot\sigma}{|\Delta_K|^{\varepsilon/n}} = \frac{\sqrt{n}\cdot\sigma}{\sqrt{m}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n}}.$$

Asymptotically $P\left(|\omega| < \frac{1}{2}\right) \to 1$ as $n \to \infty$.

Main result

ω is distributed by Gaussian with the standard deviation

$$\frac{\sqrt{n}\cdot\sigma}{|\Delta_K|^{\varepsilon/n}} = \frac{\sqrt{n}\cdot\sigma}{\sqrt{m}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n}}.$$

Asymptotically $P\left(|\omega| < \frac{1}{2}\right) \to 1$ as $n \to \infty$.
So a SCG-LWE sample

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \lambda' \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

Main result

ω is distributed by Gaussian with the standard deviation

$$\frac{\sqrt{n}\cdot\sigma}{|\Delta_K|^{\varepsilon/n}} = \frac{\sqrt{n}\cdot\sigma}{\sqrt{m}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n}}.$$

Asymptotically $P\left(|\omega| < \frac{1}{2}\right) \to 1$ as $n \to \infty$.
So a SCG-LWE sample results in

$$b_n = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle + \omega$$

Main result

ω is distributed by Gaussian with the standard deviation

$$\frac{\sqrt{n}\cdot\sigma}{|\Delta_K|^{\varepsilon/n}} = \frac{\sqrt{n}\cdot\sigma}{\sqrt{m}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n}}.$$

Asymptotically $P\left(|\omega| < \frac{1}{2}\right) \to 1$ as $n \to \infty$.
So a SCG-LWE sample results in

$$b_n = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle + \omega$$

$$\lfloor b_n \rceil = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle$$

Main result

ω is distributed by Gaussian with the standard deviation

$$\frac{\sqrt{n}\cdot\sigma}{|\Delta_K|^{\varepsilon/n}} = \frac{\sqrt{n}\cdot\sigma}{\sqrt{m}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n}}.$$

Asymptotically $P\left(|\omega| < \frac{1}{2}\right) \to 1$ as $n \to \infty$.
So a SCG-LWE sample results in

$$b_n = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle + \omega$$

$$\lfloor b_n \rceil = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle$$

n exact equations reveal the secret vector **s**.

Main result

ω is distributed by Gaussian with the standard deviation

$$\frac{\sqrt{n}\cdot\sigma}{|\Delta_K|^{\varepsilon/n}} = \frac{\sqrt{n}\cdot\sigma}{\sqrt{m}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n}}.$$

Asymptotically $P\left(|\omega| < \frac{1}{2}\right) \to 1$ as $n \to \infty$.
So a SCG-LWE sample results in

$$b_n = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle + \omega$$

$$\lfloor b_n \rceil = \langle \text{the last row of } A_{\mathbf{a}}, \mathbf{s}\rangle$$

n exact equations reveal the secret vector **s**.

The attack works for the corresponding Ring-LWE problem with

$$\sigma' = \frac{\sigma}{|\Delta|^{\varepsilon/n}}.$$

Conclusion

- No threat to the security proof of Ring-LWE.
  The standard deviation is far less than needed.

$$\sigma' = \frac{\sigma}{|\Delta|^{\varepsilon/n}} \leq \frac{1}{2\sqrt{n\log n}}.$$

Conclusion

- No threat to the security proof of Ring-LWE.
  The standard deviation is far less than needed.

$$\sigma' = \frac{\sigma}{|\Delta|^{\varepsilon/n}} \leq \frac{1}{2\sqrt{n \log n}}.$$

- SCG-LWE can simplify Ring-LWE.
  - Keep a scalar λ instead of $A_{f'(x)}$.

Conclusion

- No threat to the security proof of Ring-LWE.
  The standard deviation is far less than needed.

$$\sigma' = \frac{\sigma}{|\Delta|^{\varepsilon/n}} \le \frac{1}{2\sqrt{n \log n}}.$$

- SCG-LWE can simplify Ring-LWE.
  - Keep a scalar λ instead of $A_{f'(x)}$.
- Inaccurate choice of a scalar leads to attacks.
  - ELOS'15, CLS'15, CLS'16,
  - unified overview in Peikert'16.

Conclusion

- No threat to the security proof of Ring-LWE.

  The standard deviation is far less than needed.

$$\sigma' = \frac{\sigma}{|\Delta|^{\varepsilon/n}} \leq \frac{1}{2\sqrt{n \log n}}.$$

- SCG-LWE can simplify Ring-LWE.
  - Keep a scalar λ instead of $A_{f'(x)}$.
- Inaccurate choice of a scalar leads to attacks.
  - ELOS‘15, CLS‘15, CLS‘16,
  - unified overview in Peikert‘16.
- Hardness proof for proper scalars?

Conclusion

- No threat to the security proof of Ring-LWE.
  The standard deviation is far less than needed.

$$\sigma' = \frac{\sigma}{|\Delta|^{\varepsilon/n}} \leq \frac{1}{2\sqrt{n \log n}}.$$

- SCG-LWE can simplify Ring-LWE.
  - Keep a scalar λ instead of $A_{f'(x)}$.
- Inaccurate choice of a scalar leads to attacks.
  - ELOS‘15, CLS‘15, CLS‘16,
  - unified overview in Peikert‘16.
- Hardness proof for proper scalars?

Thank you for your attention!