

Better Algorithms for LWE and LWR

Alexandre Duc, Florian Tramèr, Serge Vaudenay

EPFL, Lausanne, Switzerland

Eurocrypt 2015, Sofia



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Many crypto primitives are based on Learning With Errors

- Trapdoor functions + IBE [Gentry et al., 2008]
- Public-key and symmetric-key cryptosystems [Regev, 2009], [Peikert, 2009], [Applebaum et al., 2009]
- FHE [Brakerski and Vaikuntanathan, 2011], [Brakerski, 2012], [Gentry et al., 2013]

Our Goal

Better understand the hardness of LWE through an algorithmic analysis, in order to propose concrete security parameters for these schemes

Many crypto primitives are based on Learning With Errors

- Trapdoor functions + IBE [Gentry et al., 2008]
- Public-key and symmetric-key cryptosystems [Regev, 2009], [Peikert, 2009], [Applebaum et al., 2009]
- FHE [Brakerski and Vaikuntanathan, 2011], [Brakerski, 2012], [Gentry et al., 2013]

Our Goal

Better understand the hardness of LWE through an algorithmic analysis, in order to propose concrete security parameters for these schemes

- Lattice reduction algorithms (LLL, BKZ, ...)

- ⇒ No precise analysis for large dimensions

- Blum-Kalai-Wasserman (BKW) Algorithm

- ⇒ Asymptotic complexity well understood

- $2^{\Theta(\frac{k}{\log k})}$ for LPN
 - $2^{\Theta(k)}$ for LWE

- ⇒ Precise algorithmic analysis

- LPN

[Blum et al., 2003], [Levieil and Fouque, 2006]
[Fossorier et al., 2006], [Bernstein and Lange, 2012]
[Guo et al., 2014], [Bogos et al., 2015]

- LWE
 - LWR

[Albrecht et al., 2013, 2014]

- Lattice reduction algorithms (LLL, BKZ, ...)

- ⇒ No precise analysis for large dimensions

- Blum-Kalai-Wasserman (BKW) Algorithm

- ⇒ Asymptotic complexity well understood

- $2^{\Theta(\frac{k}{\log k})}$ for LPN

- $2^{\Theta(k)}$ for LWE

- ⇒ Precise algorithmic analysis

- LPN

[Blum et al., 2003], [Levieil and Fouque, 2006]
[Fossorier et al., 2006], [Bernstein and Lange, 2012]
[Guo et al., 2014], [Bogos et al., 2015]

- LWE

[Albrecht et al., 2013, 2014]

- LWR

- Lattice reduction algorithms (LLL, BKZ, ...)

- ⇒ No precise analysis for large dimensions

- Blum-Kalai-Wasserman (BKW) Algorithm

- ⇒ Asymptotic complexity well understood

- $2^{\Theta\left(\frac{k}{\log k}\right)}$ for LPN

- $2^{\Theta(k)}$ for LWE

- ⇒ Precise algorithmic analysis

- LPN

[Blum et al., 2003], [Levieil and Fouque, 2006]
[Fossorier et al., 2006], [Bernstein and Lange, 2012]
[Guo et al., 2014], [Bogos et al., 2015]

- LWE

[Albrecht et al., 2013, 2014]

- LWR

- Lattice reduction algorithms (LLL, BKZ, ...)

- ⇒ No precise analysis for large dimensions

- Blum-Kalai-Wasserman (BKW) Algorithm

- ⇒ Asymptotic complexity well understood

- $2^{\Theta(\frac{k}{\log k})}$ for LPN

- $2^{\Theta(k)}$ for LWE

- ⇒ Precise algorithmic analysis

- LPN

[Blum et al., 2003], [Levieil and Fouque, 2006]

[Fossorier et al., 2006], [Bernstein and Lange, 2012]

[Guo et al., 2014], [Bogos et al., 2015]

- **LWE**

[Albrecht et al., 2013, 2014]

- **LWR**

This talk

LWE Definition

Definition (LWE Oracle)

Let k, q be positive integers. A *Learning with Errors (LWE)* oracle $\Pi_{\mathbf{s}, \chi}$ for a hidden vector $\mathbf{s} \in \mathbb{Z}_q^k$ and a probability distribution χ over \mathbb{Z}_q is an oracle returning

$$\left(\mathbf{a} \xleftarrow{U} \mathbb{Z}_q^k, \underbrace{\langle \mathbf{a}, \mathbf{s} \rangle}_{c} + \nu \right),$$

where $\nu \leftarrow \chi$.

Definition (Search-LWE)

The *Search-LWE* problem is the problem of recovering the hidden secret \mathbf{s} given n queries $(\mathbf{a}^{(j)}, c^{(j)}) \in \mathbb{Z}_q^k \times \mathbb{Z}_q$ obtained from $\Pi_{\mathbf{s}, \chi}$.

LWE Definition

Definition (LWE Oracle)

Let k, q be positive integers. A *Learning with Errors (LWE)* oracle $\Pi_{\mathbf{s}, \chi}$ for a hidden vector $\mathbf{s} \in \mathbb{Z}_q^k$ and a probability distribution χ over \mathbb{Z}_q is an oracle returning

$$\left(\mathbf{a} \xleftarrow{U} \mathbb{Z}_q^k, \underbrace{\langle \mathbf{a}, \mathbf{s} \rangle}_{\mathbf{c}} + \nu \right),$$

where $\nu \leftarrow \chi$.

Definition (Search-LWE)

The *Search-LWE* problem is the problem of recovering the hidden secret \mathbf{s} given n queries $(\mathbf{a}^{(j)}, \mathbf{c}^{(j)}) \in \mathbb{Z}_q^k \times \mathbb{Z}_q$ obtained from $\Pi_{\mathbf{s}, \chi}$.

Error Distribution(s)

Two main Gaussian error distributions appear in the literature

Definition (Rounded Gaussian Distribution

[Regev, 2009; Albrecht et al., 2013])

- Sample $x \sim \mathcal{N}(0, \sigma^2)$.
- Output $\lceil x \rceil \pmod{q} \in] -\frac{q}{2}, \frac{q}{2}]$.

Definition (Discrete Gaussian Distribution

[Regev, 2009; Brakerski et al., 2013])

$$\Pr[x] \propto \exp(-x^2/(2\sigma^2)) , \quad \text{for } x \in] -\frac{q}{2}, \frac{q}{2}] .$$

⇒ Our results apply to both distributions for practical parameters

⇒ We focus on the **discrete Gaussian distribution** for this talk

Error Distribution(s)

Two main Gaussian error distributions appear in the literature

Definition (Rounded Gaussian Distribution

[Regev, 2009; Albrecht et al., 2013])

- Sample $x \sim \mathcal{N}(0, \sigma^2)$.
- Output $\lceil x \rceil \pmod{q} \in] - \frac{q}{2}, \frac{q}{2}]$.

Definition (Discrete Gaussian Distribution

[Regev, 2009; Brakerski et al., 2013])

$$\Pr[x] \propto \exp(-x^2/(2\sigma^2)) , \quad \text{for } x \in] - \frac{q}{2}, \frac{q}{2}] .$$

⇒ Our results apply to both distributions for practical parameters

⇒ We focus on the **discrete Gaussian distribution** for this talk

Error Distribution(s)

Two main Gaussian error distributions appear in the literature

Definition (Rounded Gaussian Distribution)

[Regev, 2009; Albrecht et al., 2013]

- Sample $x \sim \mathcal{N}(0, \sigma^2)$.
- Output $\lceil x \rceil \pmod{q} \in] - \frac{q}{2}, \frac{q}{2}]$.

Definition (Discrete Gaussian Distribution)

[Regev, 2009; Brakerski et al., 2013]

$$\Pr[x] \propto \exp(-x^2/(2\sigma^2)) , \quad \text{for } x \in] - \frac{q}{2}, \frac{q}{2}] .$$

- ⇒ Our results apply to both distributions for practical parameters
- ⇒ We focus on the **discrete Gaussian distribution** for this talk

The BKW Algorithm

Reduction Phase ([Blum et al., 2003; Albrecht et al., 2013])

- In each oracle query, split \mathbf{a} into r blocks of b elements ($r \cdot b = k$)

$$\left(\begin{bmatrix} a_1 & \dots & a_b \end{bmatrix} \begin{bmatrix} a_{b+1} & \dots & a_{2b} \end{bmatrix} \dots \begin{bmatrix} a_{(r-1)b+1} & \dots & a_{rb} \end{bmatrix} \mid c \right)$$

The BKW Algorithm

Reduction Phase ([Blum et al., 2003; Albrecht et al., 2013])

- In each oracle query, split \mathbf{a} into r blocks of b elements ($r \cdot b = k$)

$$([a_1 \dots a_b] [a_{b+1} \dots a_{2b}] \dots [a_{(r-1)b+1} \dots a_{rb}] \mid c)$$

- Partition queries according to values of first block

$$\begin{array}{ccc|ccc|ccc|c} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 2 & -1 & 4 \end{bmatrix} & \begin{bmatrix} -2 & 0 & 1 \end{bmatrix} & & -1 \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -2 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -5 & 1 & -1 \end{bmatrix} & & 2 \\ \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 3 & 3 & -4 \end{bmatrix} & \begin{bmatrix} 0 & 4 & 2 \end{bmatrix} & & 0 \\ \hline \begin{bmatrix} 0 & 0 & 2 \end{bmatrix} & \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 4 & -3 \end{bmatrix} & & -5 \\ \begin{bmatrix} 0 & 0 & -2 \end{bmatrix} & \begin{bmatrix} -1 & 1 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 1 \end{bmatrix} & & 3 \\ \begin{bmatrix} 0 & 0 & -2 \end{bmatrix} & \begin{bmatrix} -2 & 5 & -5 \end{bmatrix} & \begin{bmatrix} 1 & 3 & -4 \end{bmatrix} & & 2 \end{array}$$

...

BKW reduction in \mathbb{Z}_{11}^9 , $r = 3$, $b = 3$

The BKW Algorithm

Reduction Phase ([Blum et al., 2003; Albrecht et al., 2013])

- In each oracle query, split a into r blocks of b elements ($r \cdot b = k$)

$$\left(\begin{bmatrix} a_1 & \dots & a_b \end{bmatrix} \begin{bmatrix} a_{b+1} & \dots & a_{2b} \end{bmatrix} \dots \begin{bmatrix} a_{(r-1)b+1} & \dots & a_{rb} \end{bmatrix} \mid c \right)$$

- Partition queries according to values of **first block**, and **combine**

$$\begin{array}{l}
 + \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \left[\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{array} \right] \left[\begin{array}{ccc} 2 & -1 & 4 \\ -2 & 0 & 1 \\ 3 & 3 & -4 \end{array} \right] \left[\begin{array}{ccc} -2 & 0 & 1 \\ -5 & 1 & -1 \\ 0 & 4 & 2 \end{array} \right] \left| \begin{array}{c} -1 \\ 2 \\ 0 \end{array} \right. \\
 \hline
 + \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \left[\begin{array}{ccc} 0 & 0 & 2 \\ 0 & 0 & -2 \\ 0 & 0 & -2 \end{array} \right] \left[\begin{array}{ccc} 0 & 2 & 0 \\ -1 & 1 & -3 \\ -2 & 5 & -5 \end{array} \right] \left[\begin{array}{ccc} -1 & 4 & -3 \\ 5 & 5 & 1 \\ 1 & 3 & -4 \end{array} \right] \left| \begin{array}{c} -5 \\ 3 \\ 2 \end{array} \right.
 \end{array}$$

...

BKW reduction in \mathbb{Z}_{11}^9 , $r = 3$, $b = 3$

The BKW Algorithm

Reduction Phase ([Blum et al., 2003; Albrecht et al., 2013])

- In each oracle query, split a into r blocks of b elements ($r \cdot b = k$)

$$\left(\begin{bmatrix} a_1 & \dots & a_b \end{bmatrix} \begin{bmatrix} a_{b+1} & \dots & a_{2b} \end{bmatrix} \dots \begin{bmatrix} a_{(r-1)b+1} & \dots & a_{rb} \end{bmatrix} \mid c \right)$$

- Partition queries according to values of **first block**, and **combine**

$$\begin{array}{l}
 + \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 & 4 \\ 4 & -1 & 3 \\ 5 & 2 & 0 \end{bmatrix} \begin{bmatrix} -2 & 0 & 1 \\ 3 & -1 & 2 \\ -2 & 4 & 3 \end{bmatrix} \left| \begin{array}{l} -1 \\ -3 \\ -1 \end{array} \right. \\
 \hline
 + \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 \\ -1 & 3 & -3 \\ -2 & -4 & -5 \end{bmatrix} \begin{bmatrix} -1 & 4 & -3 \\ 4 & -2 & -2 \\ 0 & -4 & 4 \end{bmatrix} \left| \begin{array}{l} -5 \\ -2 \\ -3 \end{array} \right.
 \end{array}$$

• • •

BKW reduction in \mathbb{Z}_{11}^9 , $r = 3$, $b = 3$

The BKW Algorithm

Reduction Phase ([Blum et al., 2003; Albrecht et al., 2013])

- In each oracle query, split \mathbf{a} into r blocks of b elements ($r \cdot b = k$)

$$\left(\begin{bmatrix} a_1 & \dots & a_b \end{bmatrix} \begin{bmatrix} a_{b+1} & \dots & a_{2b} \end{bmatrix} \dots \begin{bmatrix} a_{(r-1)b+1} & \dots & a_{rb} \end{bmatrix} \mid c \right)$$

- Delete the leftover query in each partition

$$\begin{array}{ccc|c} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 4 \end{bmatrix} & \begin{bmatrix} -2 & 0 & 1 \end{bmatrix} & -1 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 4 & -1 & 3 \end{bmatrix} & \begin{bmatrix} 3 & -1 & 2 \end{bmatrix} & -3 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 5 & 2 & 0 \end{bmatrix} & \begin{bmatrix} -2 & 4 & 3 \end{bmatrix} & -1 \\ \hline \begin{bmatrix} 0 & 0 & 2 \end{bmatrix} & \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 4 & -3 \end{bmatrix} & -5 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 3 & -3 \end{bmatrix} & \begin{bmatrix} 4 & -2 & -2 \end{bmatrix} & -2 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -2 & -4 & -5 \end{bmatrix} & \begin{bmatrix} 0 & -4 & 4 \end{bmatrix} & -3 \end{array}$$

...

BKW reduction in \mathbb{Z}_{11}^9 , $r = 3$, $b = 3$

The BKW Algorithm

Reduction Phase ([Blum et al., 2003; Albrecht et al., 2013])

- In each oracle query, split \mathbf{a} into r blocks of b elements ($r \cdot b = k$)

$$\left(\begin{bmatrix} a_1 & \dots & a_b \end{bmatrix} \begin{bmatrix} a_{b+1} & \dots & a_{2b} \end{bmatrix} \dots \begin{bmatrix} a_{(r-1)b+1} & \dots & a_{rb} \end{bmatrix} \mid c \right)$$

- Iterate $r - 1$ times until a single non-zero block remains

$$\begin{array}{ccc|c} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 4 & -3 \end{bmatrix} & 1 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 2 & 0 & -1 \end{bmatrix} & -2 \\ \hline \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & -4 & 0 \end{bmatrix} & 1 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 3 \end{bmatrix} & 0 \end{array}$$

...

BKW reduction in \mathbb{Z}_{11}^9 , $r = 3$, $b = 3$

The BKW Algorithm

Solving Phase ([Albrecht et al., 2013])

- Apply a last reduction to obtain queries with 1 non-zero element
- The noise now corresponds to the sum of 2^r variables drawn from χ

$$c' - \langle \mathbf{a}', \mathbf{s} \rangle = \nu_1 \pm \nu_2 \pm \cdots \pm \nu_{2^r}$$

- Guess 1 element of the secret \mathbf{s} by maximum-likelihood estimation
 - Let m denote the number of remaining queries
 - Exhaustive search through all q possibilities $\rightarrow \Theta(m \cdot q)$

The BKW Algorithm

Solving Phase ([Albrecht et al., 2013])

- Apply a last reduction to obtain queries with 1 non-zero element
- The noise now corresponds to the sum of 2^r variables drawn from χ

$$c' - \langle a', s \rangle = \nu_1 \pm \nu_2 \pm \dots \pm \nu_{2^r}$$

- Guess 1 element of the secret s by maximum-likelihood estimation
 - Let m denote the number of remaining queries
 - Exhaustive search through all q possibilities $\rightarrow \Theta(m \cdot q)$

The BKW Algorithm

Solving Phase ([Albrecht et al., 2013])

- Apply a last reduction to obtain queries with 1 non-zero element
- The noise now corresponds to the sum of 2^r variables drawn from χ

$$c' - \langle a', s \rangle = \nu_1 \pm \nu_2 \pm \dots \pm \nu_{2^r}$$

- Guess 1 element of the secret s by maximum-likelihood estimation
 - Let m denote the number of remaining queries
 - Exhaustive search through all q possibilities $\rightarrow \Theta(m \cdot q)$

The BKW Algorithm (Discrete Transforms)

Alternative Solving Phase

- Guess a block of b elements of s at once by computing a DFT
- Idea proposed by Levieil and Fouque for LPN [Levieil and Fouque, 2006]
 - Significant improvement over original BKW [Blum et al., 2003]
 - Still asymptotically $2^{\Theta(\frac{k}{\log k})}$
- Can be generalized for LWE (and LWR)
 - One reduction less \rightarrow lower noise
 - FFT algorithms $\rightarrow \Theta(m' + q^b \cdot b \cdot \log q)$

The BKW Algorithm (Discrete Transforms)

Alternative Solving Phase

- Guess a block of b elements of s at once by computing a DFT
- Idea proposed by Leveil and Fouque for LPN [Leveil and Fouque, 2006]
 - Significant improvement over original BKW [Blum et al., 2003]
 - Still asymptotically $2^{\Theta(\frac{k}{\log k})}$
- Can be generalized for LWE (and LWR)
 - One reduction less \rightarrow lower noise
 - FFT algorithms $\rightarrow \Theta(m' + q^b \cdot b \cdot \log q)$

The BKW Algorithm (Discrete Transforms)

Alternative Solving Phase

- Guess a block of b elements of s at once by computing a DFT
- Idea proposed by Leveil and Fouque for LPN [Leveil and Fouque, 2006]
 - Significant improvement over original BKW [Blum et al., 2003]
 - Still asymptotically $2^{\Theta(\frac{k}{\log k})}$
- Can be generalized for LWE (and LWR)
 - One reduction less \rightarrow lower noise
 - FFT algorithms $\rightarrow \Theta(m' + q^b \cdot b \cdot \log q)$

Could be better than
 $\Theta(m \cdot q)$ for MLE

- We improve the results of [Albrecht et al., 2013] by applying a DFT in the solving phase
 - **Remove heuristic assumptions** about sums of rounded Gaussians
 - Conceptually **simpler analysis** → closed form expression for time complexity
- **First algorithmic cryptanalysis of LWR** using similar techniques

Our Solving Phase

- After $(r-1)$ reduction rounds, we have m queries $(\mathbf{a}^{(i)}, \mathbf{c}^{(i)})$ remaining
 - \Rightarrow View the $\mathbf{a}^{(i)}$ as elements in \mathbb{Z}_q^b
 - \Rightarrow Let $\mathbf{s}' \in \mathbb{Z}_q^b$ be the secret block to recover.
 - \Rightarrow Let $\theta_q := \exp(2\pi i/q)$

- Define

$$f(\mathbf{x}) := \sum_{j=1}^m \mathbb{1}_{\{\mathbf{a}^{(j)} = \mathbf{x}\}} \theta_q^{\mathbf{c}^{(j)} \cdot \mathbf{x}}, \quad \forall \mathbf{x} \in \mathbb{Z}_q^b$$

- The DFT of f is

$$\hat{f}(\boldsymbol{\alpha}) = \sum_{j=1}^m \theta_q^{-\langle \mathbf{a}^{(j)}, \boldsymbol{\alpha} \rangle - \mathbf{c}^{(j)} \cdot \boldsymbol{\alpha}}, \quad \forall \boldsymbol{\alpha} \in \mathbb{Z}_q^b$$

- In particular

$$\hat{f}(\mathbf{s}') = \sum_{j=1}^m \theta_q^{-\langle \mathbf{a}^{(j)}, \mathbf{s}' \rangle - \mathbf{c}^{(j)} \cdot \mathbf{s}'}$$

Our Solving Phase

- After $(r-1)$ reduction rounds, we have m queries $(\mathbf{a}^{(i)}, \mathbf{c}^{(i)})$ remaining
 - \Rightarrow View the $\mathbf{a}^{(i)}$ as elements in \mathbb{Z}_q^b
 - \Rightarrow Let $\mathbf{s}' \in \mathbb{Z}_q^b$ be the secret block to recover.
 - \Rightarrow Let $\theta_q := \exp(2\pi i/q)$

- Define

$$f(\mathbf{x}) := \sum_{j=1}^m \mathbb{1}_{\{\mathbf{a}^{(j)} = \mathbf{x}\}} \theta_q^{\mathbf{c}^{(j)}}, \quad \forall \mathbf{x} \in \mathbb{Z}_q^b$$

- The DFT of f is

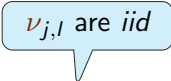
$$\hat{f}(\boldsymbol{\alpha}) = \sum_{j=1}^m \theta_q^{-\langle \mathbf{a}^{(j)}, \boldsymbol{\alpha} \rangle - \mathbf{c}^{(j)}}, \quad \forall \boldsymbol{\alpha} \in \mathbb{Z}_q^b$$

- In particular

$$\hat{f}(\mathbf{s}') = \sum_{j=1}^m \theta_q^{-\nu_{j,1} \pm \dots \pm \nu_{j,2^{r-1}}}$$

DFT Distinguisher

For the correct secret block \mathbf{s}' , we have

$$\begin{aligned}\mathbb{E} \left[\hat{f}(\mathbf{s}') \right] &= \sum_{j=1}^m \mathbb{E} \left[\theta_q^{-(\nu_{j,1} \pm \dots \pm \nu_{j,2^r-1})} \right] \\ &= \sum_{j=1}^m \mathbb{E} \left[\cos \left(\frac{2\pi}{q} \nu_{j,1} \right) + i \cdot \sin \left(\frac{2\pi}{q} \nu_{j,1} \right) \right]^{2^{r-1}}\end{aligned}$$


Lemma

For q an odd integer, let $X \sim \chi$ where χ is a discrete Gaussian over \mathbb{Z}_q with parameter σ . Let $Y \sim 2\pi X/q$. Then

$$\mathbb{E}[\cos(Y)] \geq 1 - \frac{2\pi^2\sigma^2}{q^2} \quad \text{and} \quad \mathbb{E}[\sin(Y)] = 0 .$$

Proof: Follows from Lemma 1.3 in [Banaszczyk, 1993].

For the correct secret block \mathbf{s}' , we have

$$\begin{aligned}\mathbb{E} \left[\hat{f}(\mathbf{s}') \right] &= \sum_{j=1}^m \mathbb{E} \left[\theta_q^{-\left(\nu_{j,1} \pm \dots \pm \nu_{j,2^{r-1}}\right)} \right] \\ &= \sum_{j=1}^m \mathbb{E} \left[\underbrace{\cos \left(\frac{2\pi}{q} \nu_{j,1} \right)}_{\geq 1 - 2\pi^2 \sigma^2 / q^2} + i \cdot \underbrace{\sin \left(\frac{2\pi}{q} \nu_{j,1} \right)}_0 \right]^{2^{r-1}}\end{aligned}$$

For the correct secret block \mathbf{s}' , we have

$$\begin{aligned}\mathbb{E} \left[\hat{f}(\mathbf{s}') \right] &= \sum_{j=1}^m \mathbb{E} \left[\theta_q^{-(\nu_{j,1} \pm \dots \pm \nu_{j,2^r-1})} \right] \\ &= \sum_{j=1}^m \mathbb{E} \left[\cos \left(\frac{2\pi}{q} \nu_{j,1} \right) + i \cdot \sin \left(\frac{2\pi}{q} \nu_{j,1} \right) \right]^{2^{r-1}} \\ &\geq m \cdot \left(1 - \frac{2\pi^2 \sigma^2}{q^2} \right)^{2^{r-1}}.\end{aligned}$$

For the correct secret block \mathbf{s}' , we have

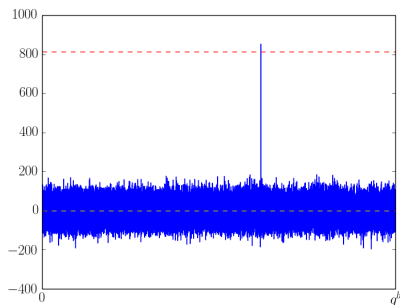
$$\begin{aligned}\mathbb{E} \left[\hat{f}(\mathbf{s}') \right] &= \sum_{j=1}^m \mathbb{E} \left[\theta_q^{-\left(\nu_{j,1} \pm \dots \pm \nu_{j,2^r-1}\right)} \right] \\ &= \sum_{j=1}^m \mathbb{E} \left[\cos \left(\frac{2\pi}{q} \nu_{j,1} \right) + i \cdot \sin \left(\frac{2\pi}{q} \nu_{j,1} \right) \right]^{2^{r-1}} \\ &\geq m \cdot \left(1 - \frac{2\pi^2 \sigma^2}{q^2} \right)^{2^{r-1}} .\end{aligned}$$

For a fixed $\alpha \neq \mathbf{s}'$, we have

$$\mathbb{E} \left[\hat{f}(\alpha) \right] = 0 .$$

Example graph of $\text{Re}(\hat{f})$, for small parameters adapted from [Regev, 2009]:

$$q = 17, \sigma = 0.85, r = 6, b = 4, m = 2^{12}$$



$$\mathbb{E} \left[\hat{f}(s') \right] \geq 811$$

$$\mathbb{E} \left[\hat{f}(\alpha) \right] = 0$$

- Algorithm: output $\operatorname{argmax}_{\alpha} \operatorname{Re}(\hat{f}(\alpha))$

- Failure Probability:

$$\Pr[\operatorname{argmax}_{\alpha} \operatorname{Re}(\hat{f}(\alpha)) \neq \mathbf{s}'] \leq q^b \cdot \exp\left(-\frac{m}{8} \cdot \left(1 - \frac{2\pi^2\sigma^2}{q^2}\right)^{2r}\right).$$

⇒ Follows from Hoeffding's inequality and a union bound

- Algorithm: output $\operatorname{argmax}_{\alpha} \operatorname{Re}(\hat{f}(\alpha))$
- Failure Probability:

$$\Pr[\operatorname{argmax}_{\alpha} \operatorname{Re}(\hat{f}(\alpha)) \neq \mathbf{s}'] \leq q^b \cdot \exp\left(-\frac{m}{8} \cdot \left(1 - \frac{2\pi^2\sigma^2}{q^2}\right)^{2^r}\right).$$

⇒ Follows from Hoeffding's inequality and a union bound

Regev's cryptosystem [Regev, 2009] with success probability 0.99.

$$q = \text{nextPrime}(k^2), \quad \sigma = O\left(\frac{q}{\sqrt{k} \log^2 k}\right)$$

k	q	$\log(\#ops)$	$\log(\#ops)$ [Albrecht et al., 2013]
64	4 099	52.62	54.85
80	6 421	63.23	65.78
96	9 221	73.72	76.75
112	12 547	85.86	87.72
128	16 411	95.03	98.67
160	25 601	115.87	120.43
224	50 177	160.34	163.76
256	65 537	178.74	185.35

- Deterministic variant of LWE
- Hardness reductions from LWE [Banerjee et al., 2012; Alwen et al., 2013]
 - ⇒ Exponential parameters or bound on oracle samples
- Many applications for PRFs [Banerjee et al., 2012; Boneh et al., 2013]

LWR Definition

Definition (LWR Oracle)

Let k and $q \geq p \geq 2$ be positive integers. A *Learning with Rounding* (LWR) oracle $\Lambda_{\mathbf{s},p}$ for a hidden vector $\mathbf{s} \in \mathbb{Z}_q^k$, $\mathbf{s} \neq \mathbf{0}$ is an oracle returning

$$\left(\mathbf{a} \xleftarrow{U} \mathbb{Z}_q^k, \underbrace{\left\lceil \left(\frac{p}{q} \right) \langle \mathbf{a}, \mathbf{s} \rangle \right\rceil}_c \right).$$

\Rightarrow For fixed \mathbf{a} , \mathbf{s} the 'errors' introduced by the oracle are deterministic

Definition (Search-LWR)

The *Search-LWR* problem is the problem of recovering the hidden secret \mathbf{s} given n queries $(\mathbf{a}^{(j)}, \mathbf{c}^{(j)}) \in \mathbb{Z}_q^k \times \mathbb{Z}_p$ obtained from $\Lambda_{\mathbf{s},p}$.

Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky
- Analysis of the **characteristic function** of the rounding errors

$$\mathbb{E} \left[e^{it\xi} \right] \text{ for } t \in \mathbb{R}, \xi = \left(\frac{p}{q} \right) \langle \mathbf{a}, \mathbf{s} \rangle - c$$

- In LWR, \mathbf{a} and ξ are **not independent!**
 - Since $\mathbf{a}^{(i)} \perp \mathbf{a}^{(j)}$ we still have $\xi^{(i)} \perp \xi^{(j)}$ for $i \neq j$
- For q prime and $p \geq 4$, we get
 - A lower bound for $\mathbb{E} \left[\hat{f}(\mathbf{s}') \right]$
 - An upper bound for $\mathbb{E} \left[\hat{f}(\alpha) \right]$ for a fixed $\alpha \neq \mathbf{s}'$

Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky
- Analysis of the **characteristic function** of the rounding errors

$$\mathbb{E} \left[e^{it\xi} \right] \text{ for } t \in \mathbb{R}, \xi = \left(\frac{p}{q} \right) \langle \mathbf{a}, \mathbf{s} \rangle - c$$

- In LWR, \mathbf{a} and ξ are **not independent!**
 - Since $\mathbf{a}^{(i)} \perp \mathbf{a}^{(j)}$ we still have $\xi^{(i)} \perp \xi^{(j)}$ for $i \neq j$
- For q prime and $p \geq 4$, we get
 - A lower bound for $\mathbb{E} \left[\hat{f}(\mathbf{s}') \right]$
 - An upper bound for $\mathbb{E} \left[\hat{f}(\alpha) \right]$ for a fixed $\alpha \neq \mathbf{s}'$

Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky
- Analysis of the **characteristic function** of the rounding errors

$$\mathbb{E} \left[e^{it\xi} \right] \text{ for } t \in \mathbb{R}, \xi = \left(\frac{p}{q} \right) \langle \mathbf{a}, \mathbf{s} \rangle - c$$

- In LWR, \mathbf{a} and ξ are **not independent!**
 - Since $\mathbf{a}^{(i)} \perp \mathbf{a}^{(j)}$ we still have $\xi^{(i)} \perp \xi^{(j)}$ for $i \neq j$
- For q prime and $p \geq 4$, we get
 - A lower bound for $\mathbb{E} \left[\hat{f}(\mathbf{s}') \right]$
 - An upper bound for $\mathbb{E} \left[\hat{f}(\alpha) \right]$ for a fixed $\alpha \neq \mathbf{s}'$

Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky
- Analysis of the **characteristic function** of the rounding errors

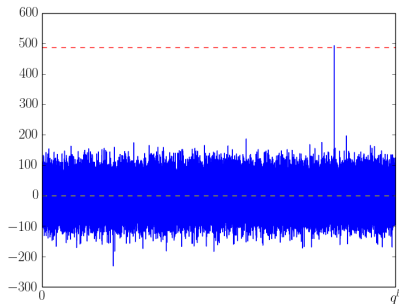
$$\mathbb{E} \left[e^{it\xi} \right] \text{ for } t \in \mathbb{R}, \xi = \left(\frac{p}{q} \right) \langle \mathbf{a}, \mathbf{s} \rangle - c$$

- In LWR, \mathbf{a} and ξ are **not independent!**
 - Since $\mathbf{a}^{(i)} \perp \mathbf{a}^{(j)}$ we still have $\xi^{(i)} \perp \xi^{(j)}$ for $i \neq j$
- For q prime and $p \geq 4$, we get
 - A lower bound for $\mathbb{E} \left[\hat{f}(\mathbf{s}') \right]$
 - An upper bound for $\mathbb{E} \left[\hat{f}(\alpha) \right]$ for a fixed $\alpha \neq \mathbf{s}'$

Results

Example graph of $\text{Re}(\hat{f})$ for small parameters adapted from [Regev, 2009] and [Alwen et al., 2013]

$$q = 17, p = 5, r = 6, b = 4, m = 2^{12}$$



$$\mathbb{E} \left[\hat{f}(s') \right] \geq 488$$

$$\mathbb{E} \left[\hat{f}(\alpha) \right] \leq 0.0003$$

- Find a better algorithm for LWR that leverages the fact that **errors are deterministic**
- Prove that LWR with **polynomial parameters** and **unlimited oracle samples** is hard
- Analyze the **heuristic independence-assumptions** used in various works on BKW for LPN and LWE