# Better Algorithms for LWE and LWR

Alexandre Duc, Florian Tramèr, Serge Vaudenay

EPFL, Lausanne, Switzerland

Eurocrypt 2015, Sofia

# LWE Applications

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

# LWE Applications

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

# Prior Work

- 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

# Prior Work

- 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

# Prior Work

- Lattice reduction algorithms (LLL, BKZ, ...)
  - $\Rightarrow$ No precise analysis for large dimensions
- Blum-Kalai-Wasserman (BKW) Algorithm
  - $\Rightarrow$ Asymptotic complexity well understood
    - $2^{\Theta\left(\frac{k}{\log k}\right)}$ for LPN
    - $2^{\Theta(k)}$ for LWE
  - $\Rightarrow$ 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

# Prior Work

- 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

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 $\chi$ over $\mathbb{Z}_q$ is an oracle returning

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

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 $\chi$ over $\mathbb{Z}_q$ is an oracle returning

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

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 $\Pi_{\mathbf{s},\chi}$.

# 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 $\lceil x \rfloor \pmod q \in ]-\frac{q}{2}, \frac{q}{2}]$.

**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 $\lceil x \rfloor \pmod{q} \in ]-\frac{q}{2}, \frac{q}{2}]$.

**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 $\lceil x \rfloor \pmod{q} \in ]-\frac{q}{2}, \frac{q}{2}]$.

**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 $\boldsymbol{a}$ 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 $\boldsymbol{a}$ into $r$ blocks of $b$ elements ($r \cdot b = k$)

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

- Partition queries according to values of first block

$$\begin{array}{ccccccccccccccc|c}
[ & 0 & 0 & 1 & ] & [ & 2 & -1 & 4 & ] & [ & -2 & 0 & 1 & ] & -1 \\
[ & 0 & 0 & 1 & ] & [ & -2 & 0 & 1 & ] & [ & -5 & 1 & -1 & ] & 2 \\
[ & 0 & 0 & -1 & ] & [ & 3 & 3 & -4 & ] & [ & 0 & 4 & 2 & ] & 0 \\
\hline
[ & 0 & 0 & 2 & ] & [ & 0 & 2 & 0 & ] & [ & -1 & 4 & -3 & ] & -5 \\
[ & 0 & 0 & -2 & ] & [ & -1 & 1 & -3 & ] & [ & 5 & 5 & 1 & ] & 3 \\
[ & 0 & 0 & -2 & ] & [ & -2 & 5 & -5 & ] & [ & 1 & 3 & -4 & ] & 2
\end{array}$$

$$\dots$$

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

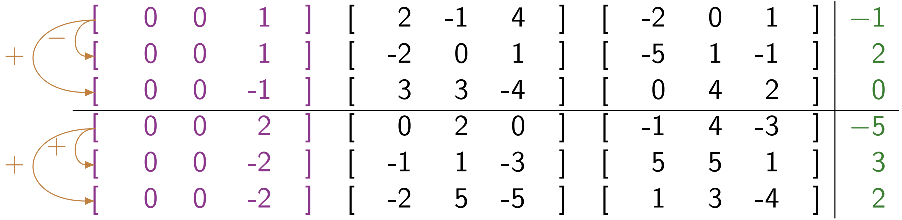
# The BKW Algorithm

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

- In each oracle query, split $\boldsymbol{a}$ 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



$$\begin{array}{cccc|c}
[\;0 & 0 & 1\;] & [\;2 & -1 & 4\;] & [\;-2 & 0 & 1\;] & -1 \\
[\;0 & 0 & 1\;] & [\;-2 & 0 & 1\;] & [\;-5 & 1 & -1\;] & 2 \\
[\;0 & 0 & -1\;] & [\;3 & 3 & -4\;] & [\;0 & 4 & 2\;] & 0 \\
\hline
[\;0 & 0 & 2\;] & [\;0 & 2 & 0\;] & [\;-1 & 4 & -3\;] & -5 \\
[\;0 & 0 & -2\;] & [\;-1 & 1 & -3\;] & [\;5 & 5 & 1\;] & 3 \\
[\;0 & 0 & -2\;] & [\;-2 & 5 & -5\;] & [\;1 & 3 & -4\;] & 2
\end{array}$$

$\dots$

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

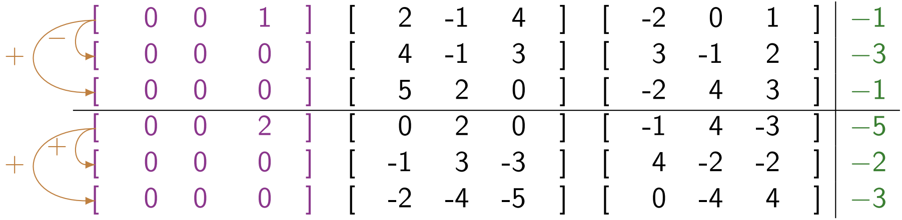
# The BKW Algorithm

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

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

$$\left( \left[ a_1 \; \ldots \; a_b \right] \left[ a_{b+1} \; \ldots \; a_{2b} \right] \; \ldots \; \left[ a_{(r-1)b+1} \; \ldots \; a_{rb} \right] \; \middle| \; c \right)$$

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



$$\begin{array}{ccc|c}
[\,0\;\;0\;\;1\,] & [\,2\;\;{-1}\;\;4\,] & [\,{-2}\;\;0\;\;1\,] & -1 \\
[\,0\;\;0\;\;0\,] & [\,4\;\;{-1}\;\;3\,] & [\,3\;\;{-1}\;\;2\,] & -3 \\
[\,0\;\;0\;\;0\,] & [\,5\;\;2\;\;0\,] & [\,{-2}\;\;4\;\;3\,] & -1 \\
\hline
[\,0\;\;0\;\;2\,] & [\,0\;\;2\;\;0\,] & [\,{-1}\;\;4\;\;{-3}\,] & -5 \\
[\,0\;\;0\;\;0\,] & [\,{-1}\;\;3\;\;{-3}\,] & [\,4\;\;{-2}\;\;{-2}\,] & -2 \\
[\,0\;\;0\;\;0\,] & [\,{-2}\;\;{-4}\;\;{-5}\,] & [\,0\;\;{-4}\;\;4\,] & -3
\end{array}$$

$\ldots$

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 ($r \cdot b = k$)

$$\left(\left[a_1 \;\ldots\; a_b\right] \left[a_{b+1} \;\ldots\; a_{2b}\right] \;\ldots\; \left[a_{(r-1)b+1} \;\ldots\; a_{rb}\right] \;\middle|\; c\right)$$

- Delete the leftover query in each partition

$$\begin{array}{ccccc ccccc ccccc|c}
\cancel{[} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{]} & \cancel{[} & \cancel{2} & \cancel{-1} & \cancel{4} & \cancel{]} & \cancel{[} & \cancel{-2} & \cancel{0} & \cancel{1} & \cancel{]} & \cancel{-1} \\
[ & 0 & 0 & 0 & ] & [ & 4 & -1 & 3 & ] & [ & 3 & -1 & 2 & ] & -3 \\
[ & 0 & 0 & 0 & ] & [ & 5 & 2 & 0 & ] & [ & -2 & 4 & 3 & ] & -1 \\
\hline
\cancel{[} & \cancel{0} & \cancel{0} & \cancel{2} & \cancel{]} & \cancel{[} & \cancel{0} & \cancel{2} & \cancel{0} & \cancel{]} & \cancel{[} & \cancel{-1} & \cancel{4} & \cancel{-3} & \cancel{]} & \cancel{-5} \\
[ & 0 & 0 & 0 & ] & [ & -1 & 3 & -3 & ] & [ & 4 & -2 & -2 & ] & -2 \\
[ & 0 & 0 & 0 & ] & [ & -2 & -4 & -5 & ] & [ & 0 & -4 & 4 & ] & -3
\end{array}$$

$$\cdots$$

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 $\boldsymbol{a}$ into $r$ blocks of $b$ elements ($r \cdot b = k$)

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

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

$$\begin{array}{ccccccccccccccc|c}
[ & 0 & 0 & 0 & ] & [ & 0 & 0 & 0 & ] & [ & -1 & 4 & -3 & ] & 1 \\
[ & 0 & 0 & 0 & ] & [ & 0 & 0 & 0 & ] & [ & 2 & 0 & -1 & ] & -2 \\
\hline
[ & 0 & 0 & 0 & ] & [ & 0 & 0 & 0 & ] & [ & 1 & -4 & 0 & ] & 1 \\
[ & 0 & 0 & 0 & ] & [ & 0 & 0 & 0 & ] & [ & -1 & -1 & 3 & ] & 0
\end{array}$$

$$\ldots$$

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 $\chi$

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

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

# 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 $\chi$

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

- 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 → $\Theta(m \cdot q)$

# 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 $\chi$

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

- 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 (Discrete Transforms)

**Alternative Solving Phase**

- Guess a block of $b$ elements of $\boldsymbol{s}$ 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 $2^{\Theta\left(\frac{k}{\log k}\right)}$

- Can be generalized for LWE (and LWR)
  - One reduction less → lower noise
  - FFT algorithms → $\Theta(m' + q^b \cdot b \cdot \log q)$

# The BKW Algorithm (Discrete Transforms)

**Alternative Solving Phase**

- Guess a block of $b$ elements of $\boldsymbol{s}$ 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 $2^{\Theta\left(\frac{k}{\log k}\right)}$

- Can be generalized for LWE (and LWR)
  - One reduction less → lower noise
  - FFT algorithms → $\Theta(m' + q^b \cdot b \cdot \log q)$

# The BKW Algorithm (Discrete Transforms)

**Alternative Solving Phase**

- Guess a block of $b$ elements of $\boldsymbol{s}$ 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 → lower noise
  - FFT algorithms → $\Theta(m' + q^b \cdot b \cdot \log q)$

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

# Our Results

- 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 $(\mathbf{a}^{(i)}, c^{(i)})$ 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

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

- The DFT of $f$ is

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

- In particular

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

# Our Solving Phase

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

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

- The DFT of $f$ is

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

- In particular

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

# DFT Distinguisher

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

$$\mathbb{E}\left[\hat{f}(\boldsymbol{s'}))\right] = \sum_{j=1}^{m} \mathbb{E}\left[\theta_q^{-(\nu_{j,1} \pm \cdots \pm \nu_{j,2^{r}-1})}\right]$$

$$= \sum_{j=1}^{m} \mathbb{E}\Big[\cos\left(\frac{2\pi}{q}\nu_{j,1}\right) + i \cdot \sin\left(\frac{2\pi}{q}\nu_{j,1}\right)\Big]^{2^{r-1}}$$

$\nu_{j,l}$ are *iid*

# DFT Distinguisher

## Lemma

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

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

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

# DFT Distinguisher

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

$$
\begin{aligned}
\mathbb{E}\left[\hat{f}(\mathbf{s'}))\right] &= \sum_{j=1}^{m} \mathbb{E}\left[\theta_q^{-(\nu_{j,1}\pm\cdots\pm\nu_{j,2^r-1})}\right] \\
&= \sum_{j=1}^{m} \mathbb{E}\Big[\underbrace{\cos\left(\frac{2\pi}{q}\nu_{j,1}\right)}_{\geq 1-2\pi^2\sigma^2/q^2} + i\cdot\underbrace{\sin\left(\frac{2\pi}{q}\nu_{j,1}\right)}_{0}\Big]^{2^{r-1}}
\end{aligned}
$$

# DFT Distinguisher

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

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

# DFT Distinguisher

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

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

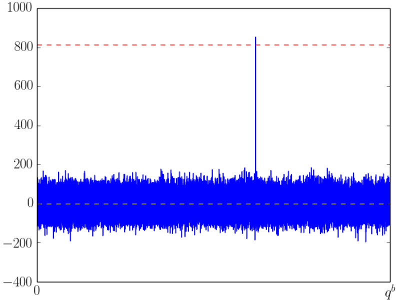
For a fixed $\boldsymbol{\alpha} \neq \boldsymbol{s'}$, we have

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

# DFT Distinguisher

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

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



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

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

# DFT Distinguisher

- Algorithm: output $\underset{\boldsymbol{\alpha}}{\texttt{argmax}}\,\mathrm{Re}(\hat{f}(\boldsymbol{\alpha}))$
- Failure Probability:

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

$\Rightarrow$ Follows from Hoeffding's inequality and a union bound

# DFT Distinguisher

- Algorithm: output $\underset{\boldsymbol{\alpha}}{\mathtt{argmax}}\,\mathrm{Re}(\hat{f}(\boldsymbol{\alpha}))$
- Failure Probability:

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

$\Rightarrow$ Follows from Hoeffding's inequality and a union bound

# LWE Results

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

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

| $k$ | $q$ | log(#ops) | log(#ops)<br>[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 |

# Learning With Roundings

- 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 $\Lambda_{\boldsymbol{s},p}$ for a hidden vector $\boldsymbol{s} \in \mathbb{Z}_q^k$, $\boldsymbol{s} \neq \mathbf{0}$ is an oracle returning

$$\left( \boldsymbol{a} \xleftarrow{U} \mathbb{Z}_q^k \, , \, \underbrace{\left\lceil \left( \frac{p}{q} \right) \langle \boldsymbol{a}, \boldsymbol{s} \rangle \right\rfloor}_{c} \right) .$$

⇒ For fixed $\boldsymbol{a}$, $\boldsymbol{s}$ the 'errors' introduced by the oracle are deterministic

## Definition (Search-LWR)

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

# Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky

- Analysis of the **characteristic function** of the rounding errors

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

- In LWR, $\boldsymbol{a}$ and $\xi$ are **not independent**!
  - Since $\boldsymbol{a}^{(i)} \perp\!\!\!\perp \boldsymbol{a}^{(j)}$ we still have $\xi^{(i)} \perp\!\!\!\perp \xi^{(j)}$ for $i \neq j$

- For $q$ prime and $p \geq 4$, we get
  - A lower bound for $\mathbb{E}\left[\hat{f}(\boldsymbol{s}')\right]$
  - An upper bound for $\mathbb{E}\left[\hat{f}(\alpha)\right]$ for a fixed $\alpha \neq \boldsymbol{s}'$

# Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky

- Analysis of the **characteristic function** of the rounding errors

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

- In LWR, $\boldsymbol{a}$ and $\xi$ are **not independent**!
  - Since $\boldsymbol{a}^{(i)} \perp\!\!\!\perp \boldsymbol{a}^{(j)}$ we still have $\xi^{(i)} \perp\!\!\!\perp \xi^{(j)}$ for $i \neq j$

- For $q$ prime and $p \geq 4$, we get
  - A lower bound for $\mathbb{E}\left[\hat{f}(\boldsymbol{s}')\right]$
  - An upper bound for $\mathbb{E}\left[\hat{f}(\alpha)\right]$ for a fixed $\alpha \neq \boldsymbol{s}'$

# Algorithm Analysis (sketch)

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

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

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

# Algorithm Analysis (sketch)

- Same algorithm as for LWE but the analysis is more tricky

- Analysis of the **characteristic function** of the rounding errors

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

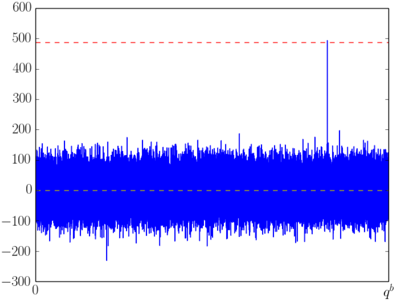
- In LWR, $\boldsymbol{a}$ and $\xi$ are **not independent**!
  - Since $\boldsymbol{a}^{(i)} \perp\!\!\!\perp \boldsymbol{a}^{(j)}$ we still have $\xi^{(i)} \perp\!\!\!\perp \xi^{(j)}$ for $i \neq j$

- For $q$ prime and $p \geq 4$, we get
  - A lower bound for $\mathbb{E}\left[\hat{f}(\boldsymbol{s}')\right]$
  - An upper bound for $\mathbb{E}\left[\hat{f}(\boldsymbol{\alpha})\right]$ for a fixed $\boldsymbol{\alpha} \neq \boldsymbol{s}'$

# Results

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

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



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

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

# Open Problems

- 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