Lattice-Based Cryptography

Chris Peikert
University of Michigan

Oxford Post-Quantum Cryptography Workshop
21 March 2019

Lattice-Based Cryptography

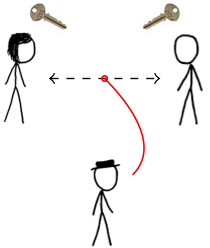
$y = g^x \bmod p$

$N = p \cdot q$

$m^e \bmod N$

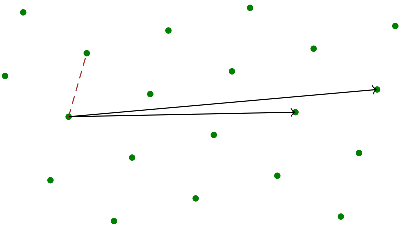
$e(g^a, g^b)$

$\Longrightarrow$

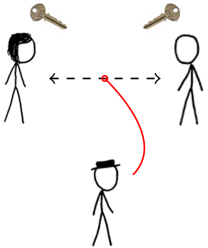


(Images courtesy xkcd.org)

Lattice-Based Cryptography

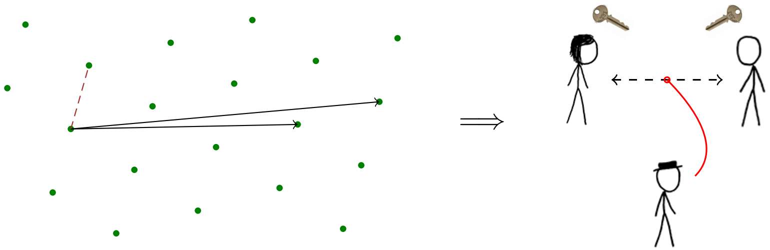


$\Longrightarrow$



(Images courtesy xkcd.org)

Lattice-Based Cryptography

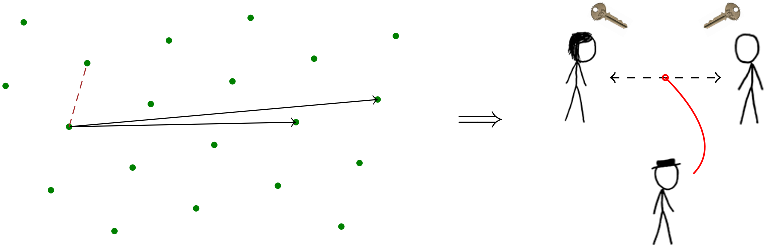


Why?

- Efficient: linear, embarrassingly parallel operations

(Images courtesy xkcd.org)

Lattice-Based Cryptography

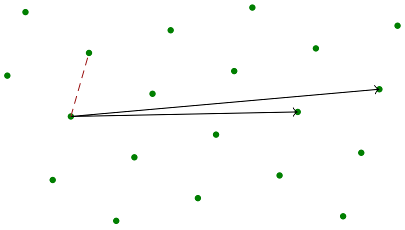


Why?

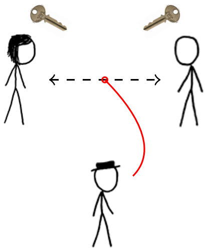
- Efficient: linear, embarrassingly parallel operations
- Appears to resist quantum attacks, contra [Shor'97]

(Images courtesy xkcd.org)

Lattice-Based Cryptography



$\Longrightarrow$

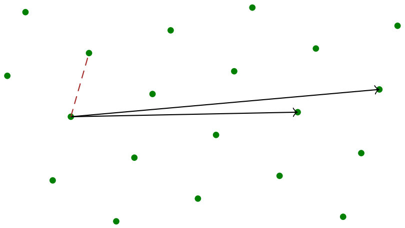


Why?

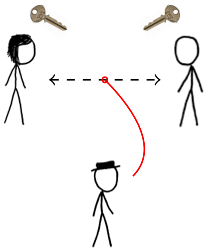
- Efficient: linear, embarrassingly parallel operations
- Appears to resist quantum attacks, contra [Shor'97]
- Security from mild worst-case assumptions

(Images courtesy xkcd.org)

Lattice-Based Cryptography



$\Longrightarrow$



Why?

- Efficient: linear, embarrassingly parallel operations
- Appears to resist quantum attacks, contra [Shor'97]
- Security from mild worst-case assumptions
- Solutions to 'holy grail' problems in crypto: FHE and related

(Images courtesy xkcd.org)

This Talk

1. Historical and mathematical background

2. Framework for lattice-based encryption/key exchange

3. Cryptanalysis, parameters, and NIST candidates

Part 1:

Background

A Brief History

1978– Rise and fall of ‘knapsack’ cryptosystems

A Brief History

1978– Rise and fall of 'knapsack' cryptosystems

1996-7 Ajtai's worst-case/average-case reduction, one-way function
& (with Dwork) public-key encryption (very inefficient)

A Brief History

1978– Rise and fall of 'knapsack' cryptosystems

1996-7 Ajtai's worst-case/average-case reduction, one-way function & (with Dwork) public-key encryption (very inefficient)

1996 NTRU efficient ring-based encryption (heuristic security)

A Brief History

1978–	Rise and fall of 'knapsack' cryptosystems
1996-7	Ajtai's worst-case/average-case reduction, one-way function & (with Dwork) public-key encryption (very inefficient)
1996	NTRU efficient ring-based encryption (heuristic security)
2002	Micciancio's ring-based one-way function with worst-case hardness (no encryption)

A Brief History

- 1978– Rise and fall of ‘knapsack’ cryptosystems
- 1996-7 Ajtai’s worst-case/average-case reduction, one-way function & (with Dwork) public-key encryption (very inefficient)
- 1996 NTRU efficient ring-based encryption (heuristic security)
- 2002 Micciancio’s ring-based one-way function with worst-case hardness (no encryption)
- 2005 Regev’s LWE: encryption with worst-case hardness (efficient-ish)

A Brief History

- 1978– Rise and fall of ‘knapsack’ cryptosystems
- 1996-7 Ajtai’s worst-case/average-case reduction, one-way function & (with Dwork) public-key encryption (very inefficient)
- 1996 NTRU efficient ring-based encryption (heuristic security)
- 2002 Micciancio’s ring-based one-way function with worst-case hardness (no encryption)
- 2005 Regev’s LWE: encryption with worst-case hardness (efficient-ish)
- 2010– Ring/Module-LWE: efficient encryption, worst-case hardness

A Brief History

1978– Rise and fall of 'knapsack' cryptosystems

1996-7 Ajtai's worst-case/average-case reduction, one-way function & (with Dwork) public-key encryption (very inefficient)

1996 NTRU efficient ring-based encryption (heuristic security)

2002 Micciancio's ring-based one-way function with worst-case hardness (no encryption)

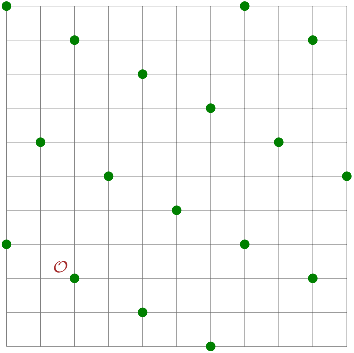
2005 Regev's LWE: encryption with worst-case hardness (efficient-ish)

2010– Ring/Module-LWE: efficient encryption, worst-case hardness

2015– Practical implementations of (Ring/Module-)LWE encryption

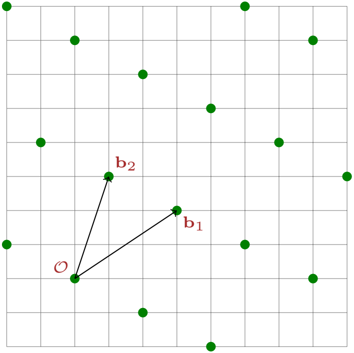
What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.



What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.

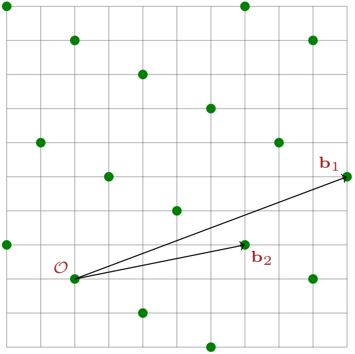


- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$:

$$\mathcal{L} \quad = \quad \sum_{i=1}^{m} (\mathbb{Z} \cdot \mathbf{b}_i)$$

What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.

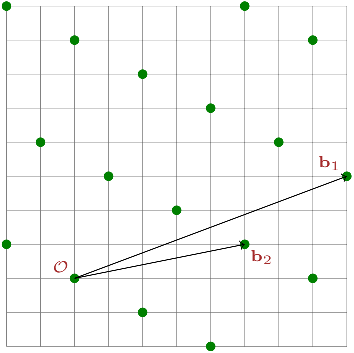


- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$:

$$\mathcal{L} \quad = \quad \sum_{i=1}^{m} (\mathbb{Z} \cdot \mathbf{b}_i)$$

What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.



- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$:

$$\mathcal{L} \quad = \quad \sum_{i=1}^{m} (\mathbb{Z} \cdot \mathbf{b}_i)$$

(Other representations as well. . .)

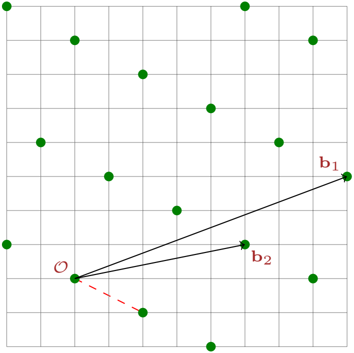
What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.

- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$:

$$\mathcal{L} \quad = \quad \sum_{i=1}^{m} (\mathbb{Z} \cdot \mathbf{b}_i)$$

(Other representations as well. . .)

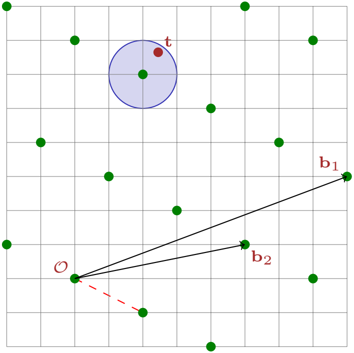


Hard Lattice Problems

- 'Find/detect short' nonzero lattice vectors.

What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.



- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$:

$$\mathcal{L} \quad = \quad \sum_{i=1}^{m} (\mathbb{Z} \cdot \mathbf{b}_i)$$

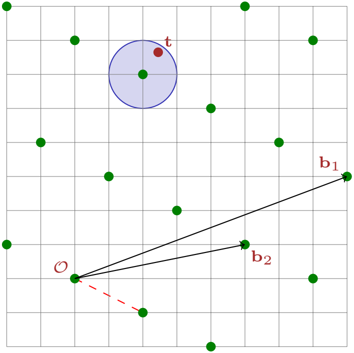
(Other representations as well. . .)

Hard Lattice Problems

- 'Find/detect short' nonzero lattice vectors.
- Decode a point 'somewhat near to' the lattice.

What's a Lattice?

- A periodic 'grid' in (subgroup of) $\mathbb{Z}^m$.



- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$:

$$\mathcal{L} \quad = \quad \sum_{i=1}^{m} (\mathbb{Z} \cdot \mathbf{b}_i)$$

(Other representations as well. . .)

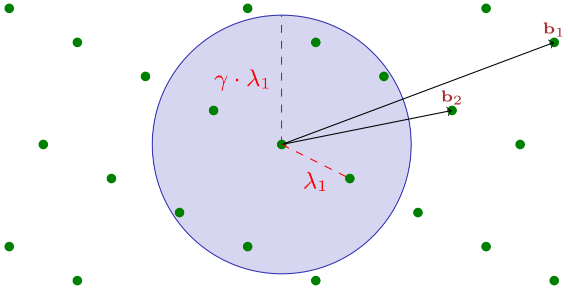
Hard Lattice Problems

- 'Find/detect short' nonzero lattice vectors.
- Decode a point 'somewhat near to' the lattice.
- Both seem to require $2^{\Omega(m)}$ time (and space).

Shortest Vector Problem: SVP_γ and GapSVP_γ

Approximation problems with factor $\gamma = \gamma(n)$:

Search: given basis $\mathbf{B}$, find nonzero $\mathbf{v} \in \mathcal{L}$ s.t. $\|\mathbf{v}\| \le \gamma \cdot \lambda_1(\mathcal{L})$.



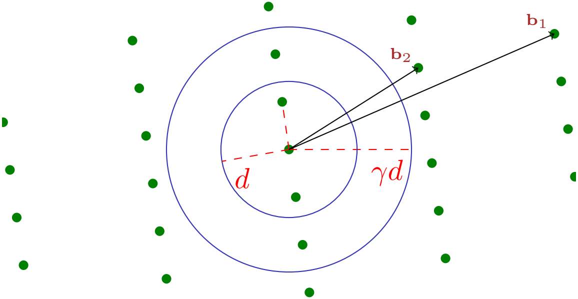
Shortest Vector Problem: SVP_γ and GapSVP_γ

Approximation problems with factor $\gamma = \gamma(n)$:

Search: given basis $\mathbf{B}$, find nonzero $\mathbf{v} \in \mathcal{L}$ s.t. $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.

Decision: given basis $\mathbf{B}$ and real d, decide whether

$$\lambda_1(\mathcal{L}) \leq d \quad \underline{\text{OR}} \quad \lambda_1(\mathcal{L}) > \gamma \cdot d.$$



Shortest Vector Problem: SVP_γ and GapSVP_γ

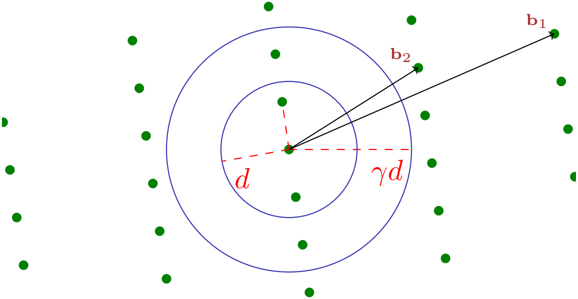
Approximation problems with factor $\gamma = \gamma(n)$:

Search: given basis $\mathbf{B}$, find nonzero $\mathbf{v} \in \mathcal{L}$ s.t. $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.

Decision: given basis $\mathbf{B}$ and real d, decide whether

$$\lambda_1(\mathcal{L}) \leq d \quad \underline{\text{OR}} \quad \lambda_1(\mathcal{L}) > \gamma \cdot d.$$

Clearly $\mathsf{GapSVP}_\gamma \leq \mathsf{SVP}_\gamma$, but the reverse direction is open!



Shortest Vector Problem: SVP_γ and GapSVP_γ

Approximation problems with factor $\gamma = \gamma(n)$:

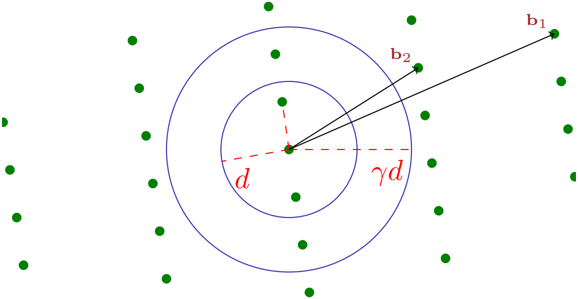
Search: given basis $\mathbf{B}$, find nonzero $\mathbf{v} \in \mathcal{L}$ s.t. $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.

Decision: given basis $\mathbf{B}$ and real d, decide whether

$$\lambda_1(\mathcal{L}) \leq d \quad \underline{\text{OR}} \quad \lambda_1(\mathcal{L}) > \gamma \cdot d.$$

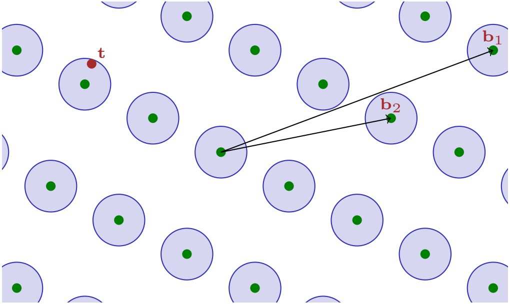
Clearly $\text{GapSVP}_\gamma \leq \text{SVP}_\gamma$, but the reverse direction is open!

Minkowski: $\min_i \|\tilde{\mathbf{b}}_i\| \leq \lambda_1(\mathcal{L}) \leq \sqrt{n} \cdot \det(\mathcal{L})^{1/n}$, but usually very loose.



Bounded-Distance Decoding (BDD)

Search: given basis $\mathbf{B}$, point $\mathbf{t}$, and real $d < \lambda_1/2$ s.t. $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq d$, find the (unique) $\mathbf{v} \in \mathcal{L}$ closest to $\mathbf{t}$.

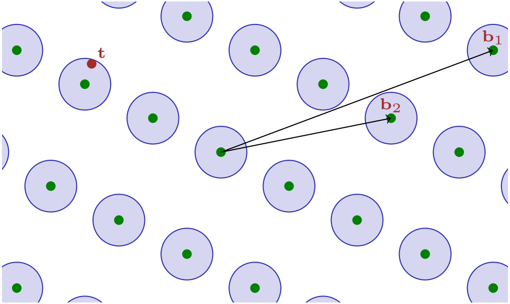


Bounded-Distance Decoding (BDD)

Search: given basis $\mathbf{B}$, point $\mathbf{t}$, and real $d < \lambda_1/2$ s.t. $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \le d$, find the (unique) $\mathbf{v} \in \mathcal{L}$ closest to $\mathbf{t}$.

Decision: given basis $\mathbf{B}$, point $\mathbf{t}$, and real d, decide whether

$$\mathrm{dist}(\mathbf{t}, \mathcal{L}) \le d \quad \underline{\text{OR}} \quad > \gamma \cdot d.$$

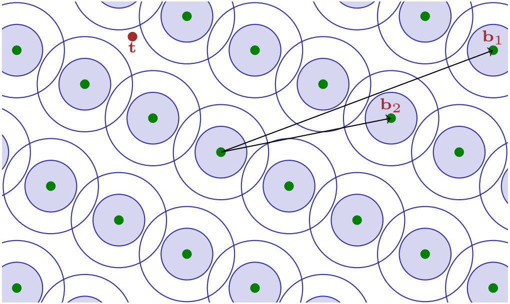


Bounded-Distance Decoding (BDD)

Search: given basis $\mathbf{B}$, point $\mathbf{t}$, and real $d < \lambda_1/2$ s.t. $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq d$, find the (unique) $\mathbf{v} \in \mathcal{L}$ closest to $\mathbf{t}$.

Decision: given basis $\mathbf{B}$, point $\mathbf{t}$, and real d, decide whether

$\mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq d$ <u>OR</u> $> \gamma \cdot d$.



A Central Hard Problem: Learning With Errors [Regev'05]

- Parameters: dimension n, modulus q, error distribution χ

A Central Hard Problem: Learning With Errors [Regev'05]

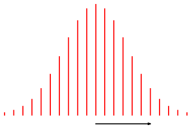
- Parameters: dimension n, modulus q, error distribution χ
- **Search:** find secret $\mathbf{s} \in \mathbb{Z}_q^n$ given many 'noisy inner products'

$$
\begin{aligned}
\mathbf{a}_1 \leftarrow \mathbb{Z}_q^n \quad &, \quad b_1 \approx \langle \mathbf{s}\,,\, \mathbf{a}_1 \rangle \bmod q \\
\mathbf{a}_2 \leftarrow \mathbb{Z}_q^n \quad &, \quad b_2 \approx \langle \mathbf{s}\,,\, \mathbf{a}_2 \rangle \bmod q \\
&\vdots \\
\mathbf{a}_m \leftarrow \mathbb{Z}_q^n \quad &, \quad b_m \approx \langle \mathbf{s}\,,\, \mathbf{a}_m \rangle \bmod q
\end{aligned}
$$

A Central Hard Problem: Learning With Errors [Regev'05]

- Parameters: dimension n, modulus q, error distribution χ
- **Search:** find <u>secret</u> $\mathbf{s} \in \mathbb{Z}_q^n$ given many 'noisy inner products'

$$
\begin{aligned}
\mathbf{a}_1 \leftarrow \mathbb{Z}_q^n \quad &, \quad b_1 = \langle \mathbf{s} \,,\, \mathbf{a}_1 \rangle + e_1 \in \mathbb{Z}_q \\
\mathbf{a}_2 \leftarrow \mathbb{Z}_q^n \quad &, \quad b_2 = \langle \mathbf{s} \,,\, \mathbf{a}_2 \rangle + e_2 \in \mathbb{Z}_q \\
&\vdots \\
\mathbf{a}_m \leftarrow \mathbb{Z}_q^n \quad &, \quad b_m = \langle \mathbf{s} \,,\, \mathbf{a}_m \rangle + e_m \in \mathbb{Z}_q
\end{aligned}
$$

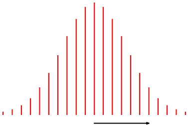


e.g. width $\sqrt{n} \ll q$, 'rate' α

A Central Hard Problem: Learning With Errors [Regev'05]

- Parameters: dimension n, modulus q, error distribution χ
- **Search:** find secret $\mathbf{s} \in \mathbb{Z}_q^n$ given many 'noisy inner products'

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \\ \end{pmatrix}}_{m} \quad , \quad \begin{pmatrix} \cdots & \mathbf{b}^t & \cdots \end{pmatrix} = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t$$

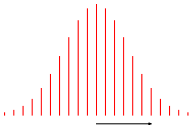


e.g. width $\sqrt{n} \ll q$, 'rate' α

A Central Hard Problem: Learning With Errors [Regev'05]

- Parameters: dimension n, modulus q, error distribution χ
- **Search:** find secret $\mathbf{s} \in \mathbb{Z}_q^n$ given many 'noisy inner products'

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \\ & & \end{pmatrix}}_{m} \quad , \quad \begin{pmatrix} \cdots & \mathbf{b}^t & \cdots \end{pmatrix} = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$$



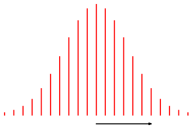
e.g. width $\sqrt{n} \ll q$, 'rate' α

- **Decision:** distinguish $(\mathbf{A}\ ,\ \mathbf{b})$ from uniform $(\mathbf{A}\ ,\ \mathbf{b})$

A Central Hard Problem: Learning With Errors [Regev'05]

- Parameters: dimension n, modulus q, error distribution χ
- **Search:** find secret $\mathbf{s} \in \mathbb{Z}_q^n$ given many 'noisy inner products'

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \\ & & \end{pmatrix}}_{m} \quad , \quad \begin{pmatrix} \cdots & \mathbf{b}^t & \cdots \end{pmatrix} = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t$$



e.g. width $\sqrt{n} \ll q$, 'rate' α

- **Decision:** distinguish $(\mathbf{A}\ ,\ \mathbf{b})$ from uniform $(\mathbf{A}\ ,\ \mathbf{b})$

LWE is Hard

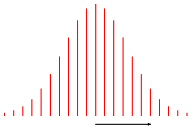
(n/α)-approx *worst case* GapSVP etc. $\leq$ search-LWE $\leq$ decision-LWE $\leq$ crypto

(quantum [R'05])   [BFKL'93,R'05,...]

A Central Hard Problem: Learning With Errors [Regev'05]

- Parameters: dimension n, modulus q, error distribution χ
- **Search:** find <u>secret</u> $\mathbf{s} \in \mathbb{Z}_q^n$ given many 'noisy inner products'

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \\ & & \end{pmatrix}}_{m}, \quad \begin{pmatrix} \cdots & \mathbf{b}^t & \cdots \end{pmatrix} = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$$



e.g. width $\sqrt{n} \ll q$, 'rate' α

- **Decision:** distinguish $(\mathbf{A}, \mathbf{b})$ from <u>uniform</u> $(\mathbf{A}, \mathbf{b})$

LWE is Hard

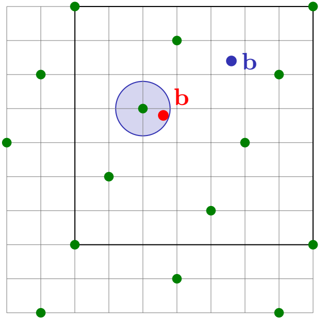
(n/α)-approx *worst case* GapSVP etc. $\leq$ search-LWE $\leq$ decision-LWE $\leq$ crypto

(quantum [R'05]) [BFKL'93,R'05,...]

- *Classical* reductions for alt. problems & params [Peikert'09,BLPRS'13]

LWE as a Lattice Problem

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \end{pmatrix}}_{m} \in \mathbb{Z}_q^{n\times m} \quad , \quad \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \quad \text{OR} \quad \mathbf{b} \leftarrow \mathbb{Z}_q^m$$



LWE as a Lattice Problem

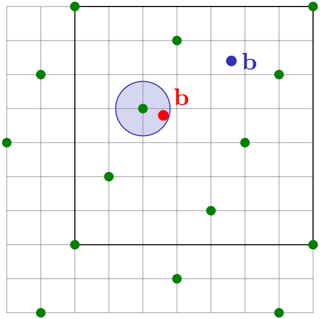
$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \end{pmatrix}}_{m} \in \mathbb{Z}_q^{n \times m} \quad , \quad \mathbf{b}^t = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t \quad \text{OR} \quad \mathbf{b} \leftarrow \mathbb{Z}_q^m$$

- Lattice interpretation:

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{z}^t \equiv \mathbf{s}^t \mathbf{A} \bmod q\}$$

Finding $\mathbf{s}, \mathbf{e}$: BDD on $\mathcal{L}(\mathbf{A})$.

Distinguishing $\mathbf{b}$ from $\mathbf{b}$: decision-BDD.



LWE as a Lattice Problem

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \end{pmatrix}}_{m} \in \mathbb{Z}_q^{n\times m} \quad , \quad \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \quad \text{OR} \quad \mathbf{b} \leftarrow \mathbb{Z}_q^m$$

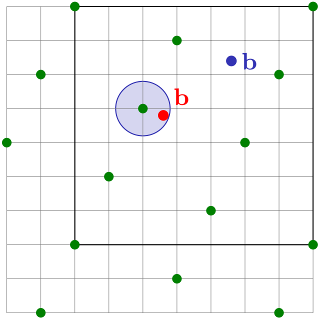
- Lattice interpretation:

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{z}^t \equiv \mathbf{s}^t\mathbf{A} \bmod q\}$$

  Finding $\mathbf{s}, \mathbf{e}$: BDD on $\mathcal{L}(\mathbf{A})$.

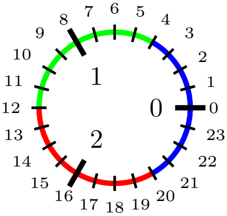
  Distinguishing $\mathbf{b}$ from $\mathbf{b}$: decision-BDD.

- WLOG, 'normal form' short $\mathbf{s} \leftarrow \chi^n$ with entries from error distribution [ACPS'09]



Learning With Rounding [BanerjeePeikertRosen'12]

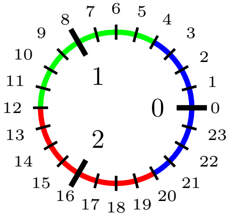
- Generate errors deterministically by rounding $\mathbb{Z}_q$ to a "sparser" subset (e.g., a subgroup).



Learning With Rounding [BanerjeePeikertRosen'12]

- Generate errors deterministically by rounding $\mathbb{Z}_q$ to a "sparser" subset (e.g., a subgroup).

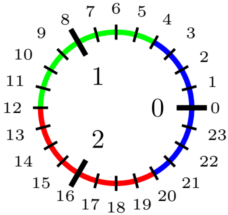
  Let $p < q$ and define $\lfloor x \rceil_p = \lfloor (p/q) \cdot x \rceil \bmod p$.



Learning With Rounding [BanerjeePeikertRosen'12]

- Generate errors deterministically by rounding $\mathbb{Z}_q$ to a "sparser" subset (e.g., a subgroup).

  Let $p < q$ and define $\lfloor x \rceil_p = \lfloor (p/q) \cdot x \rceil \bmod p$.



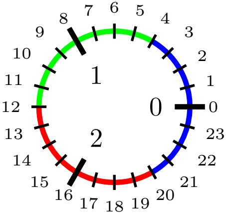
- Decision-LWR problem: for secret $\mathbf{s} \in \mathbb{Z}_q^n$, distinguish m pairs

$$\mathbf{a}_i \leftarrow \mathbb{Z}_q^n \ , \ \lfloor \langle \mathbf{s}, \mathbf{a}_i \rangle \rceil_p \in \mathbb{Z}_q^n \times \mathbb{Z}_p \quad \text{from uniform.}$$

Learning With Rounding [BanerjeePeikertRosen'12]

- Generate errors deterministically by rounding $\mathbb{Z}_q$ to a "sparser" subset (e.g., a subgroup).

  Let $p < q$ and define $\lfloor x \rceil_p = \lfloor (p/q) \cdot x \rceil \bmod p$.



- Decision-LWR problem: for secret $\mathbf{s} \in \mathbb{Z}_q^n$, distinguish m pairs

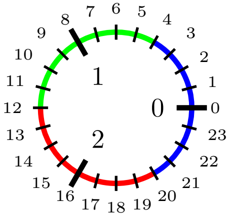
$$\mathbf{a}_i \leftarrow \mathbb{Z}_q^n \ , \ \lfloor \langle \mathbf{s}, \mathbf{a}_i \rangle \rceil_p \in \mathbb{Z}_q^n \times \mathbb{Z}_p \quad \text{from uniform.}$$

  LWE conceals low-order bits of $\langle \mathbf{s}, \mathbf{a}_i \rangle$ by adding small random error. LWR just discards those bits instead.

Learning With Rounding [BanerjeePeikertRosen'12]

- Generate errors deterministically by rounding $\mathbb{Z}_q$ to a "sparser" subset (e.g., a subgroup).

  Let $p < q$ and define $\lfloor x \rceil_p = \lfloor (p/q) \cdot x \rceil \bmod p$.



- Decision-LWR problem: for secret $\mathbf{s} \in \mathbb{Z}_q^n$, distinguish m pairs

$$\mathbf{a}_i \leftarrow \mathbb{Z}_q^n \ , \ \lfloor \langle \mathbf{s}, \mathbf{a}_i \rangle \rceil_p \in \mathbb{Z}_q^n \times \mathbb{Z}_p \quad \text{from uniform.}$$

  LWE conceals low-order bits of $\langle \mathbf{s}, \mathbf{a}_i \rangle$ by adding small random error. LWR just discards those bits instead.

- [BPR'12,AKPW'13] proves that LWE $\leq$ LWR for $q \geq p \cdot \text{poly}(m)$ ...

  ... but LWR appears hard for more aggressive parameters.

  How aggressive? Not well understood.

LWE/LWR are (Extremely) Versatile

What kinds of crypto can we do with LWE/LWR?

LWE/LWR are (Extremely) Versatile

What kinds of crypto can we do with LWE/LWR?

- ✔ Key Exchange, Public Key Encryption
- ✔ Oblivious Transfer
- ✔ Chosen Ciphertext-Secure Encryption (w/o random oracles)
- ✔ Symmetric Crypto: (Constrained & Key-Homomorphic) PRFs

LWE/LWR are (Extremely) Versatile

What kinds of crypto can we do with LWE/LWR?

- ✔ Key Exchange, Public Key Encryption
- ✔ Oblivious Transfer
- ✔ Chosen Ciphertext-Secure Encryption (w/o random oracles)
- ✔ Symmetric Crypto: (Constrained & Key-Homomorphic) PRFs

- ✔✔ Identity-Based Encryption (w/o RO)
- ✔✔ Hierarchical ID-Based Encryption (w/o RO)
- ✔✔ NIZK for any NP language

LWE/LWR are (Extremely) Versatile

What kinds of crypto can we do with LWE/LWR?

- ✔ Key Exchange, Public Key Encryption
- ✔ Oblivious Transfer
- ✔ Chosen Ciphertext-Secure Encryption (w/o random oracles)
- ✔ Symmetric Crypto: (Constrained & Key-Homomorphic) PRFs

- ✔✔ Identity-Based Encryption (w/o RO)
- ✔✔ Hierarchical ID-Based Encryption (w/o RO)
- ✔✔ NIZK for any NP language

- !!! Fully Homomorphic Encryption
- !!! Attribute-Based & Predicate Encryption for arbitrary policies

and much, much more...

Part 2:

Framework for Lattice-Based Encryption

LWE-Based Encryption/Key Ex [Regev'05,PVW'08,LPS'10,LP'11,. . .]

short $\mathbf{R} \leftarrow \chi^{k \times n}$

$\xrightarrow{\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times n}}$ (can be shared and/or expanded from a seed)

$\xrightarrow[\text{(public key)}]{\mathbf{U} \approx \mathbf{R}\mathbf{A}}$

LWE-Based Encryption/Key Ex [Regev'05,PVW'08,LPS'10,LP'11,...]

$$\xrightarrow{\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times n}}$$

(can be shared and/or expanded from a seed)

short $\mathbf{R} \leftarrow \chi^{k \times n}$

$$\xrightarrow[\text{(public key)}]{\mathbf{U} \approx \mathbf{R}\mathbf{A}}$$

$$\xleftarrow[\text{(ciphertext `preamble')}]{\mathbf{V} \approx \mathbf{A}\mathbf{S}}$$

short $\mathbf{S} \leftarrow \chi^{n \times \ell}$

msg $\mathbf{M} \in \mathbb{Z}_p^{k \times \ell}$

LWE-Based Encryption/Key Ex [Regev'05,PVW'08,LPS'10,LP'11,...]

$$\xrightarrow{\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}}$$

(can be shared and/or expanded from a seed)

short $\mathbf{R} \leftarrow \chi^{k\times n}$

$$\xrightarrow[\text{(public key)}]{\mathbf{U} \approx \mathbf{R}\mathbf{A}}$$

$$\xleftarrow[\text{(ciphertext `preamble')}]{\mathbf{V} \approx \mathbf{A}\mathbf{S}}$$

short $\mathbf{S} \leftarrow \chi^{n\times \ell}$

msg $\mathbf{M} \in \mathbb{Z}_p^{k\times \ell}$

$$\xleftarrow[\text{(ciphertext `payload')}]{\mathbf{C} \approx \mathbf{U}\mathbf{S} + \frac{q}{p}\cdot \mathbf{M}}$$

$\mathbf{R}\mathbf{V} \approx \mathbf{R}\mathbf{A}\mathbf{S}$

$\mathbf{U}\mathbf{S} \approx \mathbf{R}\mathbf{A}\mathbf{S} \in \mathbb{Z}_q^{k\times \ell}$

- - - - - - - - - - - - - - -

LWE-Based Encryption/Key Ex [Regev'05,PVW'08,LPS'10,LP'11,. . .]

$\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}$ $\longrightarrow$ (can be shared and/or expanded from a seed)

short $\mathbf{R} \leftarrow \chi^{k\times n}$

$\mathbf{U} \approx \mathbf{RA}$ $\longrightarrow$
(public key)

$\mathbf{V} \approx \mathbf{AS}$ $\longleftarrow$
(ciphertext 'preamble')

short $\mathbf{S} \leftarrow \chi^{n\times \ell}$
msg $\mathbf{M} \in \mathbb{Z}_p^{k\times \ell}$

$\mathbf{C} \approx \mathbf{US} + \frac{q}{p} \cdot \mathbf{M}$ $\longleftarrow$
(ciphertext 'payload')

$\mathbf{RV} \approx \mathbf{RAS}$

$\mathbf{US} \approx \mathbf{RAS} \in \mathbb{Z}_q^{k\times \ell}$

$(\mathbf{A}, \mathbf{U}, \mathbf{V}, \mathbf{C})$

LWE-Based Encryption/Key Ex [Regev'05,PVW'08,LPS'10,LP'11,. . .]

$$\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times n} \longrightarrow$$

(can be shared and/or expanded from a seed)

short $\mathbf{R} \leftarrow \chi^{k \times n}$

$$\xrightarrow{\mathbf{U} \approx \mathbf{RA}}$$

(public key)

$$\xleftarrow{\mathbf{V} \approx \mathbf{AS}}$$

(ciphertext 'preamble')

short $\mathbf{S} \leftarrow \chi^{n \times \ell}$

msg $\mathbf{M} \in \mathbb{Z}_p^{k \times \ell}$

$\mathbf{RV} \approx \mathbf{RAS}$

$$\xleftarrow{\mathbf{C} \approx \mathbf{US} + \frac{q}{p} \cdot \mathbf{M}}$$

(ciphertext 'payload')

$\mathbf{US} \approx \mathbf{RAS} \in \mathbb{Z}_q^{k \times \ell}$

$(\mathbf{A}, \mathbf{U}, \mathbf{V}, \mathbf{C})$

by decision-LWE

LWE-Based Encryption/Key Ex [Regev'05,PVW'08,LPS'10,LP'11,...]

$\xrightarrow{\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}}$ (can be shared and/or expanded from a seed)

short $\mathbf{R} \leftarrow \chi^{k\times n}$ $\quad \xrightarrow[\text{(public key)}]{\mathbf{U} \approx \mathbf{RA}}$

$\xleftarrow[\text{(ciphertext `preamble')}]{\mathbf{V} \approx \mathbf{AS}}$ short $\mathbf{S} \leftarrow \chi^{n\times \ell}$

msg $\mathbf{M} \in \mathbb{Z}_p^{k\times \ell}$

$\mathbf{RV} \approx \mathbf{RAS}$ $\quad \xleftarrow[\text{(ciphertext `payload')}]{\mathbf{C} \approx \mathbf{US} + \frac{q}{p} \cdot \mathbf{M}}$ $\quad \mathbf{US} \approx \mathbf{RAS} \in \mathbb{Z}_q^{k\times \ell}$

- - - - - - - - - - - - - -

$(\mathbf{A}, \mathbf{U}, \mathbf{V}, \mathbf{C})$

by decision-LWE

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

   Rounding makes keys/ciphertexts smaller; security is less understood.

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

   Rounding makes keys/ciphertexts smaller; security is less understood.

4. How large can/should errors be?

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

   Rounding makes keys/ciphertexts smaller; security is less understood.

4. How large can/should errors be?

   - ⋆ All else being equal, larger $|\text{errors}|/q \implies$ more security.

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

   Rounding makes keys/ciphertexts smaller; security is less understood.

4. How large can/should errors be?
   - ⋆ All else being equal, larger $|\text{errors}|/q \implies$ more security.
   - ⋆ But need entries of

     $$\mathbf{RE} - \mathbf{E}'\mathbf{S} + \mathbf{E}''$$

     to have magnitudes $< \frac{q}{2p}$, with high probability. So $q > p|\text{errors}|^2$.

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

   Rounding makes keys/ciphertexts smaller; security is less understood.

4. How large can/should errors be?
   - ⋆ All else being equal, larger $|\text{errors}|/q \Longrightarrow$ more security.
   - ⋆ But need entries of

     $$\mathbf{R}\mathbf{E} - \mathbf{E}'\mathbf{S} + \mathbf{E}''$$

     to have magnitudes $< \frac{q}{2p}$, with high probability. So $q > p|\text{errors}|^2$.

5. What is an acceptable decryption failure probability?

Design Considerations

1. System as shown is only CPA secure. Good for ephemeral key-ex, but needs a Fujisaki–Okamoto-like transform for CCA-secure KEM.

   An active area of research; mostly orthogonal to other design aspects.

2. Share $\mathbf{A}$ across many public keys?

   May allow (expensive) preprocessing, making it easier to break many public keys at once.

3. Use random errors, or deterministic rounding?

   Rounding makes keys/ciphertexts smaller; security is less understood.

4. How large can/should errors be?

   - ⋆ All else being equal, larger $|\text{errors}|/q \Longrightarrow$ more security.
   - ⋆ But need entries of

     $$\mathbf{RE} - \mathbf{E}'\mathbf{S} + \mathbf{E}''$$

     to have magnitudes $< \frac{q}{2p}$, with high probability. So $q > p|\text{errors}|^2$.

5. What is an acceptable decryption failure probability?

   Failures can leak secret; address 'large-error' ciphertexts [DVV'18].

Rings for Efficiency

- Matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times n}, \mathbf{R}, \mathbf{S}$ etc. were over the mod-q integer ring $\mathbb{Z}_q$. So sizes and computations grow quadratically (at least).

Rings for Efficiency

- Matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times n}, \mathbf{R}, \mathbf{S}$ etc. were over the mod-q integer ring $\mathbb{Z}_q$. So sizes and computations grow quadratically (at least).
- À la NTRU, instead use lower-dim matrices over a polynomial ring R_q. E.g., $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for power-of-two d (the $2d$th cyclotomic).

Rings for Efficiency

- Matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times n}, \mathbf{R}, \mathbf{S}$ etc. were over the mod-q integer ring $\mathbb{Z}_q$. So sizes and computations grow quadratically (at least).
- À la NTRU, instead use lower-dim matrices over a polynomial ring R_q. E.g., $R_q = \mathbb{Z}_q[X]/(X^d+1)$ for power-of-two d (the $2d$th cyclotomic).
- Extreme $n = 1$ is Ring-LWE/LWR [LPR'10]: for secret $s \in R_q$, pairs

$$a_i \leftarrow R_q \ , \ b_i \approx s \cdot a_i \in R_q .$$

Rings for Efficiency

- Matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times n}, \mathbf{R}, \mathbf{S}$ etc. were over the mod-q integer ring $\mathbb{Z}_q$. So sizes and computations grow quadratically (at least).
- À la NTRU, instead use lower-dim matrices over a polynomial ring R_q. E.g., $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for power-of-two d (the $2d$th cyclotomic).
- Extreme $n = 1$ is Ring-LWE/LWR [LPR'10]: for secret $s \in R_q$, pairs

$$a_i \leftarrow R_q \ , \ b_i \approx s \cdot a_i \in R_q .$$

- Intermediate $n \geq 2$ is Module-LWE/LWR [BGV'12,LS'15]. E.g., for secret $\mathbf{s} = (s_1, s_2) \in R_q^2$,

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \in R_q^{2 \times 2} \ , \ \mathbf{b} \approx \mathbf{s}\mathbf{A} \in R_q^2 \quad \text{from uniform.}$$

Rings for Efficiency

- Matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times n}, \mathbf{R}, \mathbf{S}$ etc. were over the mod-q integer ring $\mathbb{Z}_q$. So sizes and computations grow quadratically (at least).
- À la NTRU, instead use lower-dim matrices over a polynomial ring R_q. E.g., $R_q = \mathbb{Z}_q[X]/(X^d+1)$ for power-of-two d (the $2d$th cyclotomic).
- Extreme $n = 1$ is Ring-LWE/LWR [LPR'10]: for secret $s \in R_q$, pairs

$$a_i \leftarrow R_q \ , \ b_i \approx s \cdot a_i \in R_q .$$

- Intermediate $n \geq 2$ is Module-LWE/LWR [BGV'12,LS'15]. E.g., for secret $\mathbf{s} = (s_1, s_2) \in R_q^2$,

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \in R_q^{2\times 2} \ , \ \mathbf{b} \approx \mathbf{s}\mathbf{A} \in R_q^2 \quad \text{from uniform.}$$

- Sizes and computations can now grow only (quasi-)linearly in total dimension, thanks to FFT-like techniques.

Rings for Efficiency

- Matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times n}, \mathbf{R}, \mathbf{S}$ etc. were over the mod-q integer ring $\mathbb{Z}_q$. So sizes and computations grow quadratically (at least).
- À la NTRU, instead use lower-dim matrices over a polynomial ring R_q. E.g., $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for power-of-two d (the $2d$th cyclotomic).
- Extreme $n = 1$ is Ring-LWE/LWR [LPR'10]: for secret $s \in R_q$, pairs

$$a_i \leftarrow R_q \ , \ b_i \approx s \cdot a_i \in R_q .$$

- Intermediate $n \geq 2$ is Module-LWE/LWR [BGV'12,LS'15]. E.g., for secret $\mathbf{s} = (s_1, s_2) \in R_q^2$,

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \in R_q^{2 \times 2} \ , \ \mathbf{b} \approx \mathbf{s}\mathbf{A} \in R_q^2 \quad \text{from uniform.}$$

- Sizes and computations can now grow only (quasi-)linearly in total dimension, thanks to FFT-like techniques.

  Also (weaker) worst-case hardness theorems based on ideal lattices.

NTRU [HoffsteinPipherSilverman'96,...]

- Ring-LWE public keys (a, b) satisfy the inhomogeneous relation

$$a \cdot s - b \approx 0 \in R_q.$$

NTRU [HoffsteinPipherSilverman'96,. . .]

- Ring-LWE public keys (a, b) satisfy the inhomogeneous relation

$$a \cdot s - b \approx 0 \in R_q.$$

- NTRU is more extreme: public key $a = r \cdot s^{-1} \in R_q$ for short r, s, satisfying the homogeneous relation

$$a \cdot s \approx 0.$$

NTRU [HoffsteinPipherSilverman'96,...]

- Ring-LWE public keys (a, b) satisfy the inhomogeneous relation

$$a \cdot s - b \approx 0 \in R_q.$$

- NTRU is more extreme: public key $a = r \cdot s^{-1} \in R_q$ for short r, s, satisfying the homogeneous relation

$$a \cdot s \approx 0.$$

- Encryption is similar: choose short t and send $c \approx t \cdot a + \frac{q}{p} \cdot m \in R_q$.

  (Just one ring element!)

  Decryption:

$$c \cdot s \approx t \cdot a \cdot s + \frac{q}{p} \cdot m \cdot s \approx \frac{q}{p} \cdot m \cdot s,$$

  from which we can recover m.

Part 3:

Cryptanalysis, Parameters, and NIST Candidates

Lattice Attacks

- Standard approach: given $[\mathbf{A} \mid \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}]$, find the (unique mod $\pm$) 'unusually short' vector $(\mathbf{s}, \mathbf{e}, 1)$ in the lattice

$$\mathcal{L} = \{\mathbf{x} : [\mathbf{A} \mid -\mathbf{I} \mid -\mathbf{b}] \cdot \mathbf{x} = \mathbf{0}\}.$$

Lattice Attacks

- Standard approach: given $[\mathbf{A} \mid \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}]$, find the (unique mod $\pm$) 'unusually short' vector $(\mathbf{s}, \mathbf{e}, 1)$ in the lattice

$$\mathcal{L} = \{\mathbf{x} : [\mathbf{A} \mid -\mathbf{I} \mid -\mathbf{b}] \cdot \mathbf{x} = \mathbf{0}\}.$$

Core-SVP Methodology

- Use Block Korkin-Zolotarev (BKZ) with large enough block size b to succeed. Conservatively lower-bound the cost by a single exact-SVP computations in dimension b. (BKZ actually makes several SVP calls.)

Lattice Attacks

- Standard approach: given $[\mathbf{A} \mid \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}]$, find the (unique mod $\pm$) 'unusually short' vector $(\mathbf{s}, \mathbf{e}, 1)$ in the lattice

$$\mathcal{L} = \{\mathbf{x} : [\mathbf{A} \mid -\mathbf{I} \mid -\mathbf{b}] \cdot \mathbf{x} = \mathbf{0}\}.$$

Core-SVP Methodology

- Use Block Korkin-Zolotarev (BKZ) with large enough block size b to succeed. Conservatively lower-bound the cost by a single exact-SVP computations in dimension b. (BKZ actually makes several SVP calls.)
- E.g., best known classical SVP runtime is heuristically $2^{0.292b+o(b)}$, with significant $o(b)$ term and $2^{\Omega(b)}$ memory (which are ignored).

Lattice Attacks

- Standard approach: given $[\mathbf{A} \mid \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}]$, find the (unique mod $\pm$) 'unusually short' vector $(\mathbf{s}, \mathbf{e}, 1)$ in the lattice

$$\mathcal{L} = \{\mathbf{x} : [\mathbf{A} \mid -\mathbf{I} \mid -\mathbf{b}] \cdot \mathbf{x} = \mathbf{0}\}.$$

Core-SVP Methodology

- Use Block Korkin-Zolotarev (BKZ) with large enough block size b to succeed. Conservatively lower-bound the cost by a single exact-SVP computations in dimension b. (BKZ actually makes several SVP calls.)
- E.g., best known classical SVP runtime is heuristically $2^{0.292b+o(b)}$, with significant $o(b)$ term and $2^{\Omega(b)}$ memory (which are ignored).

Exploit Ring Structure?

- To date, we have only trivial $O(d)$-factor speedups for Ring/Module-LWE over d-dimensional rings. (NTRU? Stay tuned. . .)

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^{S}$ pairs, where $S = |\text{Support}(\chi)|$ is the number of possible integer error values.

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^S$ pairs, where $S = |\text{Support}(\chi)|$ is the number of possible integer error values.

  (For Ring-LWE/NTRU, the needed number is only $\approx n^{S-1}$.)

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^S$ pairs, where $S = |\text{Support}(\chi)|$ is the number of possible integer error values.

  (For Ring-LWE/NTRU, the needed number is only $\approx n^{S-1}$.)

  Number was reduced somewhat using Gröbner bases [ACFP'14].

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^{S}$ pairs, where $S = |\text{Support}(\chi)|$ is the number of possible integer error values.

  (For Ring-LWE/NTRU, the needed number is only $\approx n^{S-1}$.)

  Number was reduced somewhat using Gröbner bases [ACFP'14].

- This suggests a potential risk of very small (rounding) errors, e.g., $\{0, \pm 1\}$ as in NTRU, NTRU Prime, LAC, ThreeBears—although they provide few pairs.

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^{S}$ pairs, where $S = |\mathsf{Support}(\chi)|$ is the number of possible integer error values.

  (For Ring-LWE/NTRU, the needed number is only $\approx n^{S-1}$.)

  Number was reduced somewhat using Gröbner bases [ACFP'14].

- This suggests a potential risk of very small (rounding) errors, e.g., $\{0, \pm 1\}$ as in NTRU, NTRU Prime, LAC, ThreeBears—although they provide few pairs.

  (Small errors are the source of their relatively small keys/ciphertexts.)

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^{S}$ pairs, where $S = |\text{Support}(\chi)|$ is the number of possible integer error values.

  (For Ring-LWE/NTRU, the needed number is only $\approx n^{S-1}$.)

  Number was reduced somewhat using Gröbner bases [ACFP'14].

- This suggests a potential risk of very small (rounding) errors, e.g., $\{0, \pm 1\}$ as in NTRU, NTRU Prime, LAC, ThreeBears—although they provide few pairs.

  (Small errors are the source of their relatively small keys/ciphertexts.)

- FrodoKEM, Kyber, NewHope, SABER use relatively larger errors, at the cost of larger keys/ciphertexts.

Combinatorial/Algebraic Attacks

Arora-Ge'11

- Solves LWE in $\approx n^{S\omega}$ time given $\approx n^S$ pairs, where $S = |\mathsf{Support}(\chi)|$ is the number of possible integer error values.

  (For Ring-LWE/NTRU, the needed number is only $\approx n^{S-1}$.)

  Number was reduced somewhat using Gröbner bases [ACFP'14].

- This suggests a potential risk of very small (rounding) errors, e.g., $\{0, \pm 1\}$ as in NTRU, NTRU Prime, LAC, ThreeBears—although they provide few pairs.

  (Small errors are the source of their relatively small keys/ciphertexts.)

- FrodoKEM, Kyber, NewHope, SABER use relatively larger errors, at the cost of larger keys/ciphertexts.

  (Indeed, FrodoKEM's error distributions even conform to a nontrivial worst-case/average-case reduction.)

NTRU Lattice Attacks

- For NTRU key $a = r \cdot s^{-1} \in R_q$, homogeneous relation $a \cdot s \approx 0 \in R_q$ means there are d 'unusually short' planted vectors $(r \cdot X^i, s \cdot X^i)$ in the $2d$-dimensional NTRU lattice.

NTRU Lattice Attacks

- For NTRU key $a = r \cdot s^{-1} \in R_q$, homogeneous relation $a \cdot s \approx 0 \in R_q$ means there are d ‘unusually short’ planted vectors $(r \cdot X^i, s \cdot X^i)$ in the $2d$-dimensional NTRU lattice.

- [KirchnerFouque’16] noticed that this structure can significantly speed up standard lattice attacks, based on the size of the ‘unusual’ gap.

  E.g., they easily broke proposed ‘stretched’ FHE parameters, but ‘ordinary’ parameters are so far unaffected.

NTRU Lattice Attacks

- For NTRU key $a = r \cdot s^{-1} \in R_q$, homogeneous relation $a \cdot s \approx 0 \in R_q$ means there are d 'unusually short' planted vectors $(r \cdot X^i, s \cdot X^i)$ in the $2d$-dimensional NTRU lattice.
- [KirchnerFouque'16] noticed that this structure can significantly speed up standard lattice attacks, based on the size of the 'unusual' gap.

  E.g., they easily broke proposed 'stretched' FHE parameters, but 'ordinary' parameters are so far unaffected.
- These (standard) attacks subsumed all prior ones against NTRU whose effectiveness had been attributed to the existence of subrings/homomorphisms.

NTRU Lattice Attacks

- For NTRU key $a = r \cdot s^{-1} \in R_q$, homogeneous relation $a \cdot s \approx 0 \in R_q$ means there are d 'unusually short' planted vectors $(r \cdot X^i, s \cdot X^i)$ in the $2d$-dimensional NTRU lattice.
- [KirchnerFouque'16] noticed that this structure can significantly speed up standard lattice attacks, based on the size of the 'unusual' gap.

  E.g., they easily broke proposed 'stretched' FHE parameters, but 'ordinary' parameters are so far unaffected.
- These (standard) attacks subsumed all prior ones against NTRU whose effectiveness had been attributed to the existence of subrings/homomorphisms.
- This suggests a potential risk of homogeneity and NTRU lattices—regardless of choice of ring.

NTRU Lattice Attacks

- For NTRU key $a = r \cdot s^{-1} \in R_q$, homogeneous relation $a \cdot s \approx 0 \in R_q$ means there are d 'unusually short' planted vectors $(r \cdot X^i, s \cdot X^i)$ in the $2d$-dimensional NTRU lattice.
- [KirchnerFouque'16] noticed that this structure can significantly speed up standard lattice attacks, based on the size of the 'unusual' gap.

  E.g., they easily broke proposed 'stretched' FHE parameters, but 'ordinary' parameters are so far unaffected.
- These (standard) attacks subsumed all prior ones against NTRU whose effectiveness had been attributed to the existence of subrings/homomorphisms.
- This suggests a potential risk of homogeneity and NTRU lattices—regardless of choice of ring.
- By contrast, BDD problems like (Ring-/Module-)LWE plant a unique shortest vector, which [KirchnerFouque'16] explicitly recommend.

Conclusions

- Lattice-based PKE/KEM all work very similarly at heart, but there is a huge space of design choices and trade-offs.

Conclusions

- Lattice-based PKE/KEM all work very similarly at heart, but there is a huge space of design choices and trade-offs.
- Key issues: balance the risk/efficiency trade-offs inherent in:
  - randomized versus deterministic rounding,
  - size of errors,
  - decryption failures,
  - ring structure and problem rank over the ring,
  - BDD/LWE versus non-unique-SVP/NTRU,
  - and much more.

Conclusions

- Lattice-based PKE/KEM all work very similarly at heart, but there is a huge space of design choices and trade-offs.
- Key issues: balance the risk/efficiency trade-offs inherent in:
  - randomized versus deterministic rounding,
  - size of errors,
  - decryption failures,
  - ring structure and problem rank over the ring,
  - BDD/LWE versus non-unique-SVP/NTRU,
  - and much more.
- There are many great questions to investigate!

Conclusions

- Lattice-based PKE/KEM all work very similarly at heart, but there is a huge space of design choices and trade-offs.
- Key issues: balance the risk/efficiency trade-offs inherent in:
  - randomized versus deterministic rounding,
  - size of errors,
  - decryption failures,
  - ring structure and problem rank over the ring,
  - BDD/LWE versus non-unique-SVP/NTRU,
  - and much more.
- There are many great questions to investigate!

Thanks!