Fundamentals of Lattice-Based Cryptography

Chris Peikert
University of Michigan

2nd Crypto Innovation School
Shanghai, China
13 December 2019
Talk Outline

1. Lattices and hard problems

2. The SIS and LWE problems; basic applications

3. Using rings for efficiency
Today’s Cryptography (e.g., RSA, Diffie-Hellman)

- Conjectured-hard problems: factor $N = P \cdot Q$, compute discrete logs

$N =$

\[
\begin{array}{c}
1606221687090904406512585584569433331615827658775597032991663 \\
1326451405320056545967263583507984286802756201383768089567669
\end{array}
\]

$g, y = g^x \in G$
Today’s Cryptography (e.g., RSA, Diffie-Hellman)

- Conjectured-hard problems: factor $N = P \cdot Q$, compute discrete logs
- Shor’s quantum algorithm:

\[
N = 21305750140972822779 \\
67336009072353225107 \\
5886421620325802176 \\
55802658737520126407 \\
22059995071405557278 \\
967027854563351343547
\]

\[
P = 16062216870909044065 \\
12585584569433331615 \\
827658775597032991663
\]

\[
Q = 13264514053200565459 \\
67263583507984286802 \\
756201383768089567669
\]
Today’s Cryptography (e.g., RSA, Diffie-Hellman)

- Conjectured-hard problems: factor $N = P \cdot Q$, compute discrete logs
- Shor’s quantum algorithm:

\[
N = 21305750140972822279 \\
67336009072353225107 \\
58864221620325802176 \\
5580268737520126407 \\
2205999507140557278 \\
967027854563351343547
\]

\[
P = \\
16062216870909044065 \\
1258558456943331615 \\
827658775597032991663
\]

\[
Q = \\
13264514053200565459 \\
67263583507984286802 \\
756201383768089567669
\]

\[
g, y = g^X \in G
\]
Lattice-Based Cryptography

\[ y = g^x \mod p \]
\[ m^e \mod N \]
\[ N = p \cdot q \]
\[ e(g^a, g^b) \]

Advantages
▶ Appears resistant to quantum attacks
▶ Simple description and implementation
▶ Efficient: linear, highly parallelizable
▶ Security from worst-case assumptions

\[ \text{Images courtesy xkcd.org} \]
Lattice-Based Cryptography

\[ N = p \cdot q \]
\[ y = g^x \mod p \]
\[ m^e \mod N \]
\[ e(g^a, g^b) \]

Advantages

▶ Appears resistant to quantum attacks
▶ Simple description and implementation
▶ Efficient: linear, highly parallelizable
▶ Security from worst-case assumptions

(Images courtesy xkcd.org)
Lattice-Based Cryptography

\[ N = p \cdot q \]
\[ y = g^x \mod p \]
\[ m^e \mod N \]
\[ e(g^a, g^b) \]
\[ \Rightarrow \]

Advantages

▶ Appears resistant to quantum attacks

(Images courtesy xkcd.org)
Lattice-Based Cryptography

\[ N = p \cdot q \]
\[ y = g^x \mod p \]
\[ m^e \mod N = e(g^a, g^b) \]

\[ \Rightarrow \]

Advantages

▶ Appears resistant to quantum attacks
▶ Simple description and implementation

(Images courtesy xkcd.org)
Lattice-Based Cryptography

\[
N = p \cdot q \\
y = g^x \mod p \\
m^e \mod N \\
e(g^a, g^b) \Rightarrow
\]

Advantages

▶ Appears resistant to quantum attacks
▶ **Simple** description and implementation
▶ **Efficient**: linear, highly parallelizable

(Images courtesy xkcd.org)
Lattice-Based Cryptography

\[ N = p \cdot q \]
\[ y = g^x \mod p \]
\[ m^e \mod N \]
\[ e(a, b) \Rightarrow \]

Advantages

- Appears resistant to quantum attacks
- **Simple** description and implementation
- **Efficient**: linear, highly parallelizable
- **Security from worst-case assumptions** [Ajtai96, ...]

(Images courtesy xkcd.org)
Part 1:

Lattices and Hard Problems
Lattices

- An (integer) **lattice** is a subgroup $\mathcal{L} \subseteq \mathbb{Z}^m$. (Looks like a periodic “grid.”)

Conjectured Hard Problems

- Find ‘relatively short’ (nonzero) lattice vector(s): SVP $\gamma$, SIVP $\gamma$.
- Estimate geometric quantities of the lattice: minimum distance $\lambda_1$, successive minima $\lambda_i$, covering radius $\mu$, ...
Lattices

- An (integer) lattice is a subgroup $\mathcal{L} \subseteq \mathbb{Z}^m$. (Looks like a periodic “grid.”)

- Has a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ of linearly independent vectors:

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

Today, $k = m$ always: “full rank.”
Lattices

- An (integer) lattice is a subgroup $\mathcal{L} \subseteq \mathbb{Z}^m$. (Looks like a periodic “grid.”)

- Has a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ of linearly independent vectors:

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

Today, $k = m$ always: “full rank.”
Lattices

- An (integer) lattice is a subgroup $\mathcal{L} \subseteq \mathbb{Z}^m$. (Looks like a periodic “grid.”)

- Has a basis $B = \{b_1, \ldots, b_k\}$ of linearly independent vectors:

  $$\mathcal{L} = \sum_{i=1}^{k} (\mathbb{Z} \cdot b_i)$$

Today, $k = m$ always: “full rank.”

(Other representations as well...)

![Graphical representation of a lattice]

$\mathcal{O}$ $b_1$ $b_2$
Lattices

- An (integer) lattice is a subgroup $\mathcal{L} \subseteq \mathbb{Z}^m$. (Looks like a periodic “grid.”)

- Has a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ of linearly independent vectors:

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

Today, $k = m$ always: “full rank.”

(Other representations as well...)

Conjectured Hard Problems

- Find ‘relatively short’ (nonzero) lattice vector(s): SVP$_{\gamma}$, SIVP$_{\gamma}$
Lattices

▶ An (integer) lattice is a subgroup $\mathcal{L} \subseteq \mathbb{Z}^m$. (Looks like a periodic “grid.”)

▶ Has a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ of linearly independent vectors:

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

Today, $k = m$ always: “full rank.”

(Other representations as well...)

Conjectured Hard Problems

▶ Find ‘relatively short’ (nonzero) lattice vector(s): SVP$_\gamma$, SIVP$_\gamma$

▶ Estimate geometric quantities of the lattice: minimum distance $\lambda_1$, successive minima $\lambda_i$, covering radius $\mu$, ...
Complexity (for the Worst Case)

GapSVP\(\gamma\)

- Given (a basis of) an \(m\)-dim lattice \(L\) and some \(d > 0\), distinguish
  \[\lambda_1(L) \leq d\quad \text{FROM} \quad \lambda_1(L) > \gamma(m) \cdot d\]
Complexity (for the Worst Case)

**GapSVP**

- Given (a basis of) an $m$-dim lattice $\mathcal{L}$ and some $d > 0$, distinguish
  \[ \lambda_1(\mathcal{L}) \leq d \quad \text{FROM} \quad \lambda_1(\mathcal{L}) > \gamma(m) \cdot d \]
- Becomes easier for larger $\gamma(m)$:

  \[
  \gamma(m) = 2^{(\log m)^{1-\epsilon}} \quad \text{NP-hard}^{*} \quad [Ajt98, \ldots]
  \]
  \[
  \sqrt{m} \in \text{coNP}^{*} \quad [GG98, AR05]
  \]
  \[
  \sim m \in \text{P}^{*} \quad [LLL82, Sch87]
  \]

- Similar status for other problems like SIVP$_\gamma$, \ldots
Complexity (for the Worst Case)

GapSVP_\gamma

- Given (a basis of) an m-dim lattice \( \mathcal{L} \) and some \( d > 0 \), distinguish

\[
\lambda_1(\mathcal{L}) \leq d \quad \text{FROM} \quad \lambda_1(\mathcal{L}) > \gamma(m) \cdot d
\]

- Becomes easier for larger \( \gamma(m) \):

\[
\gamma = 2^{(\log m)^{1-\epsilon}} \quad \sqrt{m} \quad \gtrsim m \quad 2^{\sim m}
\]

<table>
<thead>
<tr>
<th>NP-hard*</th>
<th>( \in ) coNP</th>
<th>crypto</th>
<th>( \in ) P</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Ajt98,\ldots]</td>
<td>[GG98,AR05]</td>
<td>[Ajt96,\ldots]</td>
<td>[LLL82,Sch87]</td>
</tr>
</tbody>
</table>
Complexity (for the Worst Case)

**GapSVP}_γ**

- Given (a basis of) an \( m \)-dim lattice \( \mathcal{L} \) and some \( d > 0 \), distinguish
  \[
  \lambda_1(\mathcal{L}) \leq d \quad \text{FROM} \quad \lambda_1(\mathcal{L}) > \gamma(m) \cdot d
  \]

- Becomes easier for larger \( \gamma(m) \):
  \[
  \gamma = 2^{(\log m)^{1-\epsilon}} \quad \sqrt{m} \quad \gtrsim m \quad 2^{-m}
  \]
  - NP-hard* \[\text{[Ajt98,\ldots]}\]
  - \( \in \) coNP \[\text{[GG98,AR05]}\]
  - crypto \[\text{[Ajt96,\ldots]}\]
  - \( \in \) P \[\text{[LLL82,Sch87]}\]

- For \( \gamma = \text{poly}(m) \), fastest algorithm: \( 2^m \) time & space \[\text{[AKS01,MV10,\ldots]}\]
Complexity (for the Worst Case)

**GapSVP\(\gamma\)**

- Given (a basis of) an \(m\)-dim lattice \(\mathcal{L}\) and some \(d > 0\), distinguish
  \[\lambda_1(\mathcal{L}) \leq d\] FROM \[\lambda_1(\mathcal{L}) > \gamma(m) \cdot d\]

- Becomes easier for larger \(\gamma(m)\):
  \[\gamma = 2^{(\log m)^{1-\epsilon}}\]
  \[\sqrt{m}\]
  \[\gtrsim m\]
  \[2^{\sim m}\]
  \[
  \begin{align*}
  \text{NP-hard*} & \quad \in \text{coNP} & \quad \text{crypto} & \quad \in \text{P}
  \end{align*}
  \[
  \text{[Ajt98,\ldots]} & \quad \text{[GG98,AR05]} & \quad \text{[Ajt96,\ldots]} & \quad \text{[LLL82,Sch87]}
  \]

- For \(\gamma = \text{poly}(m)\), fastest algorithm: \(2^m\) time & space \([\text{AKS01,MV10,\ldots}]\)

- Similar status for other problems like SIVP\(\gamma\), \ldots
Part 2:

SIS/LWE and Basic Applications
A Hard Problem: Short Integer Solution [Ajtai’96]

- Fix a dimension $n$ and modulus $q$ (e.g., $q \approx n^2$).
  Let $\mathbb{Z}_q^n = n$-dimensional integer vectors modulo $q$. 
  
  SIS: given many uniform $a_i$, find 'short' nonzero $z$ s.t.
  
  Collision-Resistant Hash Function
  
  Define $f_A : \{0, 1\}^m \rightarrow \mathbb{Z}_q^n$ for any $m > n \lg q$ as $f_A(x) = Ax$.
  
  Collision $x, x' \in \{0, 1\}^m$ where $Ax = Ax'$ yields a short (nonzero) solution $z = x - x' \in \{0, \pm 1\}^m$. 

A Hard Problem: Short Integer Solution  [Ajtai’96]

- Fix a dimension $n$ and modulus $q$ (e.g., $q \approx n^2$).
  Let $\mathbb{Z}_q^n = n$-dimensional integer vectors modulo $q$.

$$
\begin{pmatrix}
\mathbf{a}_1 \\
\vdots
\end{pmatrix} \; \begin{pmatrix}
\mathbf{a}_2 \\
\vdots
\end{pmatrix} \; \cdots \; \begin{pmatrix}
\mathbf{a}_m \\
\vdots
\end{pmatrix} \; \in \mathbb{Z}_q^n
$$
A Hard Problem: Short Integer Solution [Ajtai’96]

- Fix a dimension $n$ and modulus $q$ (e.g., $q \approx n^2$).
  Let $\mathbb{Z}_q^n = n$-dimensional integer vectors modulo $q$.

- **SIS**: given many uniform $a_i$, find nontrivial $z_1, \ldots, z_m \in \{0, \pm1\}$ s.t.

\[
z_1 \cdot \begin{pmatrix} a_1 \end{pmatrix} + z_2 \cdot \begin{pmatrix} a_2 \end{pmatrix} + \cdots + z_m \cdot \begin{pmatrix} a_m \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix} \in \mathbb{Z}_q^n
\]
Fix a dimension $n$ and modulus $q$ (e.g., $q \approx n^2$).
Let $\mathbb{Z}_q^n = n$-dimensional integer vectors modulo $q$.

**SIS**: given many uniform $a_i$, find ‘short’ nonzero $z$ s.t.

\[
\begin{pmatrix}
\vdots & A & \vdots \\
\end{pmatrix}
\begin{pmatrix}
z \\
\end{pmatrix}
= 0 \in \mathbb{Z}_q^n
\]
A Hard Problem: Short Integer Solution  [Ajtai’96]

- Fix a dimension $n$ and modulus $q$ (e.g., $q \approx n^2$).
  Let $\mathbb{Z}_q^n = n$-dimensional integer vectors modulo $q$.
- **SIS**: given many uniform $a_i$, find ‘short’ nonzero $z$ s.t.

\[
\begin{pmatrix}
\vdots & A & \vdots \\
\end{pmatrix}
\begin{pmatrix}
z \\
\end{pmatrix}
= 0 \in \mathbb{Z}_q^n
\]

Collision-Resistant Hash Function

- Define $f_A : \{0, 1\}^m \rightarrow \mathbb{Z}_q^n$ for any $m > n \lg q$ as

  \[
f_A(x) = Ax.
\]
A Hard Problem: Short Integer Solution  [Ajtai’96]

- Fix a dimension \( n \) and modulus \( q \) (e.g., \( q \approx n^2 \)).
  Let \( \mathbb{Z}_q^n = n \)-dimensional integer vectors modulo \( q \).

- **SIS**: given many uniform \( a_i \), find ‘short’ nonzero \( z \) s.t.

\[
\begin{pmatrix}
\vdots & A & \vdots \\
\end{pmatrix}
\begin{pmatrix}
z \\
\end{pmatrix}
= 0 \in \mathbb{Z}_q^n
\]

Collision-Resistant Hash Function

- Define \( f_A : \{0, 1\}^m \rightarrow \mathbb{Z}_q^n \) for any \( m > n \lg q \) as

\[
f_A(x) = Ax.
\]

- Collision \( x, x' \in \{0, 1\}^m \) where \( Ax = Ax' \) …
A Hard Problem: Short Integer Solution  [Ajtai’96]

- Fix a dimension $n$ and modulus $q$ (e.g., $q \approx n^2$).
  Let $\mathbb{Z}_q^n = n$-dimensional integer vectors modulo $q$.

- **SIS**: given many uniform $a_i$, find ‘short’ nonzero $z$ s.t.

\[
\begin{pmatrix}
\cdots & A & \cdots
\end{pmatrix}
\begin{pmatrix}
z
\end{pmatrix}
= 0 \in \mathbb{Z}_q^n
\]

Collision-Resistant Hash Function

- Define $f_A : \{0, 1\}^m \rightarrow \mathbb{Z}_q^n$ for any $m > n \lg q$ as

\[
f_A(x) = Ax.
\]

- Collision $x, x' \in \{0, 1\}^m$ where $Ax = Ax' \ldots$

  ...yields a short (nonzero) solution $z = x - x' \in \{0, \pm 1\}^m$.  

Cool! (but what does this have to do with lattices?)
Cool!

- Matrix \( A = (a_1, \ldots, a_m) \in \mathbb{Z}_q^{n \times m} \):

\[
L^\perp(A) = \{ z \in \mathbb{Z}^m : Az = 0 \}
\]
Matrix $A = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbb{Z}_q^{n \times m}$:

$$\mathcal{L}^\perp(A) = \{ \mathbf{z} \in \mathbb{Z}^m : A\mathbf{z} = 0 \}$$
Matrix $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbb{Z}_q^{n \times m}$:

$$\mathcal{L}^\perp(\mathbf{A}) = \{ \mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = 0 \}$$

'Short' solutions $\mathbf{z}$ lie in the area of the circle.
Cool!

- Matrix $A = (a_1, \ldots, a_m) \in \mathbb{Z}_q^{n \times m}$:

  $$\mathcal{L}^\perp(A) = \{z \in \mathbb{Z}^m : Az = 0\}$$

- ‘Short’ solutions $z$ lie in $O(0,q)$

Worst-Case/Average-Case Connection [Ajtai96, …]

Finding ‘short’ ($\|z\| \leq \beta \ll q$) nonzero $z \in \mathcal{L}^\perp(A)$
(for uniformly random $A \in \mathbb{Z}_q^{n \times m}$)

$\Downarrow$

solving $\text{GapSVP}_{\beta\sqrt{n}}$ and $\text{SIVP}_{\beta\sqrt{n}}$ on any $n$-dim lattice
Application: Digital Signatures [GentryPeikertVaikuntanathan'08]

- Generate uniform $vk = A$ with ‘trapdoor’ $sk = T$. [Ajtai’99,...,MP’12]
Application: Digital Signatures [GentryPeikertVaikuntanathan’08]

- Generate uniform \( vk = A \) with ‘trapdoor’ \( sk = T \). [Ajtai’99, . . . , MP’12]
- Sign\((T, \mu)\): use \( T \) to sample a short \( x \in \mathbb{Z}^m \) s.t. \( Ax = H(\mu) \in \mathbb{Z}^n \).
Application: Digital Signatures [GentryPeikertVaikuntanathan’08]

- Generate uniform \( vk = A \) with ‘trapdoor’ \( sk = T \). [Ajtai’99,…,MP’12]
- \( \text{Sign}(T, \mu) \): use \( T \) to sample a short \( x \in \mathbb{Z}^m \) s.t. \( Ax = H(\mu) \in \mathbb{Z}_q^n \). Draw \( x \) from a (Gaussian) distribution, which reveals nothing about \( T \):

\[
\text{Verify}(A, \mu, x): \text{check that } Ax = H(\mu) \text{ and } x \text{ is sufficiently short.}
\]

\[
\text{Security: forging a signature for a new message } \mu^* \text{ requires finding a short } x^* \text{ s.t. } Ax^* = H(\mu^*). \text{ This is SIS!}
\]
Application: Digital Signatures [GentryPeikertVaikuntanathan'08]

- Generate uniform $vk = A$ with ‘trapdoor’ $sk = T$.  [Ajtai’99,…,MP’12]
- Sign$(T, \mu)$: use $T$ to sample a short $x \in \mathbb{Z}^m$ s.t. $Ax = H(\mu) \in \mathbb{Z}_q^n$. Draw $x$ from a (Gaussian) distribution, which reveals nothing about $T$:

- Verify$(A, \mu, x)$: check that $Ax = H(\mu)$ and $x$ is sufficiently short.
Application: Digital Signatures [GentryPeikertVaikuntanathan'08]

- Generate uniform $vk = A$ with ‘trapdoor’ $sk = T$. [Ajtai’99, . . . , MP’12]

- Sign($T, \mu$): use $T$ to sample a short $x \in \mathbb{Z}^m$ s.t. $Ax = H(\mu) \in \mathbb{Z}_q^n$. Draw $x$ from a (Gaussian) distribution, which reveals nothing about $T$:

- Verify($A, \mu, x$): check that $Ax = H(\mu)$ and $x$ is sufficiently short.

- Security: forging a signature for a new message $\mu^*$ requires finding a short $x^*$ s.t. $Ax^* = H(\mu^*)$. This is SIS!
Gaussian Sampling over a (Shifted) Lattice

- Sample \( x \) s.t. \( Ax = u \) given any ‘short’ basis \( T \): \( \max \|t_i\| \leq \text{std dev} \)
  - Output distribution **leaks no information** about secret basis \( T \)!
Gaussian Sampling over a (Shifted) Lattice

- Sample \( x \) s.t. \( Ax = u \) given any ‘short’ basis \( T \): \( \max \|t_i\| \leq \) std dev
  - Output distribution leaks no information about secret basis \( T \)!

- “Nearest-plane” algorithm with randomized rounding [Klein’00,GPV’08]
Gaussian Sampling over a (Shifted) Lattice

- Sample \( x \) s.t. \( Ax = u \) given any ‘short’ basis \( T \): \( \max \| t_i \| \leq \text{std dev} \)
  - Output distribution leaks no information about secret basis \( T \)!

- “Nearest-plane” algorithm with randomized rounding [Klein’00, GPV’08]
Gaussian Sampling over a (Shifted) Lattice

- Sample \( \mathbf{x} \) s.t. \( \mathbf{A}\mathbf{x} = \mathbf{u} \) given any ‘short’ basis \( \mathbf{T} \): \( \max \|\mathbf{t}_i\| \leq \text{std dev} \)
  - Output distribution leaks no information about secret basis \( \mathbf{T} \)!

- “Nearest-plane” algorithm with randomized rounding [Klein’00,GPV’08]
Gaussian Sampling over a (Shifted) Lattice

- Sample $x$ s.t. $Ax = u$ given any ‘short’ basis $T$: $\max \|t_i\| \leq \text{std dev}$
  - Output distribution leaks no information about secret basis $T$!

- “Nearest-plane” algorithm with randomized rounding [Klein’00,GPV’08]
Gaussian Sampling over a (Shifted) Lattice

- Sample \( x \) s.t. \( Ax = u \) given any ‘short’ basis \( T \): \( \max \| t_i \| \leq \text{std dev} \)
  - Output distribution leaks no information about secret basis \( T \)!

- “Nearest-plane” algorithm with randomized rounding [Klein’00,GPV’08]

Proof idea: \( D_{L_u^\perp} \) (plane) depends (essentially) only on \( \text{dist}(O, \text{plane}) \); not affected by shift within plane. So rounding with that probability produces that distribution.
Another Hard Problem: Learning With Errors \cite{Regev'05}

- Parameters: dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$

Search: find secret $s \in \mathbb{Z}_q^n$ given many 'noisy inner products' $\sqrt{n} \leq \text{std dev} \ll q$, 'rate' $\alpha$

Decision: distinguish $(A, b)$ from uniform $(A, b)$

$LWE$ is hard $(\frac{n}{\alpha})$-approx worst case $\text{GapSVP, SIVP} \leq \text{(quantum [R'05]) search-LWE} \leq \text{[BFKL'93,R'05,. . .]}$ decision-LWE $\leq \text{crypto}$

- Also fully classical reductions, for worse params \cite{Peikert'09,BLPRS'13}
- Also a direct worst-case $\leq$ decision-LWE (quantum) reduction \cite{PRS'17}
Another Hard Problem: Learning With Errors [Regev’05]

- Parameters: dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$
- **Search**: find secret $s \in \mathbb{Z}_q^n$ given many ‘noisy inner products’

\[
\begin{align*}
\mathbf{a}_1 & \leftarrow \mathbb{Z}_q^n, \quad b_1 \approx \langle s, \mathbf{a}_1 \rangle \mod q \\
\mathbf{a}_2 & \leftarrow \mathbb{Z}_q^n, \quad b_2 \approx \langle s, \mathbf{a}_2 \rangle \mod q \\
& \vdots
\end{align*}
\]
Another Hard Problem: Learning With Errors  [Regev’05]

- Parameters: dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$
- **Search:** find secret $s \in \mathbb{Z}_q^n$ given many ‘noisy inner products’

$$a_1 \leftarrow \mathbb{Z}_q^n, \quad b_1 = \langle s, a_1 \rangle + e_1 \in \mathbb{Z}_q$$
$$a_2 \leftarrow \mathbb{Z}_q^n, \quad b_2 = \langle s, a_2 \rangle + e_2 \in \mathbb{Z}_q$$
$$\vdots$$

$\sqrt{n} \leq \text{std dev} \ll q$, ‘rate’ $\alpha$
Another Hard Problem: Learning With Errors [Regev’05]

- Parameters: dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$
- **Search**: find secret $s \in \mathbb{Z}_q^n$ given many ‘noisy inner products’

\[
\begin{pmatrix}
\cdots & A & \cdots
\end{pmatrix},
\begin{pmatrix}
\cdots & b^t & \cdots
\end{pmatrix} \approx s^t A
\]

\[
\sqrt{n} \leq \text{std dev} \ll q, \text{‘rate’ } \alpha
\]
Another Hard Problem: Learning With Errors  [Regev’05]

- **Parameters:** dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$

- **Search:** find secret $s \in \mathbb{Z}_q^n$ given many ‘noisy inner products’

\[
\begin{pmatrix}
\cdots & A & \cdots \\
\end{pmatrix},
\begin{pmatrix}
\cdots & b^t & \cdots \\
\end{pmatrix} \approx s^t A
\]

\[
\sqrt{n} \leq \text{std dev} \ll q, \text{ ‘rate’ } \alpha
\]

- **Decision:** distinguish $(A, b)$ from uniform $(A, b)$
Another Hard Problem: Learning With Errors \[\text{[Regev'05]}\]

- **Parameters**: dimension \(n\), modulus \(q = \text{poly}(n)\), error distribution \(\chi\)

- **Search**: find secret \(s \in \mathbb{Z}_q^n\) given many ‘noisy inner products’

\[
\begin{pmatrix}
\cdots & A & \cdots \\
\cdots & b^t & \cdots
\end{pmatrix} \approx s^t A
\]

\[\sqrt{n} \leq \text{std dev} \ll q, \text{ ‘rate’ } \alpha\]

- **Decision**: distinguish \((A, b)\) from uniform \((A, b)\)

**LWE is Hard**

\[
\frac{n}{\alpha}\text{-approx worst case } \text{GapSVP, SIVP} \lessgtr \text{search-LWE} \lessgtr \text{decision-LWE} \lessgtr \text{crypto}
\]

\[\text{quantum [R'05]} \quad \text{[BFKL'93,R'05,...]}\]
Another Hard Problem: Learning With Errors  [Regev’05]

- **Parameters:** dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$
- **Search:** find secret $s \in \mathbb{Z}_q^n$ given many ‘noisy inner products’

\[
\begin{pmatrix}
\cdots & A & \cdots \\
\end{pmatrix}, \quad \begin{pmatrix}
\cdots & b^t & \cdots \\
\end{pmatrix} \approx s^t A
\]

\[
\sqrt{n} \leq \text{std dev} \ll q, \text{ ‘rate’ } \alpha
\]

- **Decision:** distinguish $(A, b)$ from uniform $(A, b)$

LWE is Hard

\[(n/\alpha)-\text{approx worst case} \quad \text{GapSVP, SIVP} \quad \leq \text{search-LWE} \quad \leq \text{decision-LWE} \quad \leq \text{crypto}\]

\[
\uparrow \quad \uparrow \quad \uparrow
\]

(quantum [R’05]) [BFKL’93,R’05,...]

- Also fully classical reductions, for worse params [Peikert’09,BLPRS’13]
Another Hard Problem: Learning With Errors  [Regev’05]

- **Parameters:** dimension $n$, modulus $q = \text{poly}(n)$, error distribution $\chi$

- **Search:** find secret $s \in \mathbb{Z}_q^n$ given many ‘noisy inner products’

\[
\begin{pmatrix}
\cdots & A & \cdots \\
\end{pmatrix}, \quad \begin{pmatrix}
\cdots & b^t & \cdots \\
\end{pmatrix} \approx s^t A
\]

\[
\sqrt{n} \leq \text{std dev} \ll q, \text{ ‘rate’ } \alpha
\]

- **Decision:** distinguish $(A, b)$ from uniform $(A, b)$

**LWE is Hard**

\[
(n/\alpha)\text{-approx worst case} \leq \text{search-LWE} \leq \text{decision-LWE} \leq \text{crypto}
\]

\[
\text{GapSVP, SIVP} \xrightarrow{\text{quantum [R'05]}} \text{search-LWE} \xrightarrow{\text{quantum}} \text{decision-LWE}
\]

- Also fully *classical* reductions, for worse params [Peikert’09, BLPRS’13]
- Also a direct worst-case $\leq \text{decision-LWE}$ (quantum) reduction [PRS’17]
LWE is Versatile

What kinds of crypto can we construct from LWE?
LWE is Versatile
What kinds of crypto can we construct from LWE?

- ✔ Key Exchange/Public Key Encryption
- ✔ Oblivious Transfer
- ✔ Actively Secure Encryption (w/o random oracles)
- ✔ (Constrained) PRFs
LWE is Versatile

What kinds of crypto can we construct from LWE?

- Key Exchange/Public Key Encryption
- Oblivious Transfer
- Actively Secure Encryption (w/o random oracles)
- (Constrained) PRFs
- Identity-Based Encryption (w/ RO)
- Hierarchical ID-Based Encryption (w/o RO)
- NIZK for NP (w/o RO)
LWE is Versatile

What kinds of crypto can we construct from LWE?

- Key Exchange/Public Key Encryption
- Oblivious Transfer
- Actively Secure Encryption (w/o random oracles)
- (Constrained) PRFs
- Identity-Based Encryption (w/ RO)
- Hierarchical ID-Based Encryption (w/o RO)
- NIZK for NP (w/o RO)
- Fully Homomorphic Encryption
- Attribute-Based/Predicate Encryption for arbitrary policies
  and much, much more...
Public-Key Cryptosystem from LWE [Regev'05, GPV'08]

short $x$

$A \leftarrow \mathbb{Z}_q^{n \times m}$
Public-Key Cryptosystem from LWE \cite{Regev'05,GPV'08}

short $x$

\[ A \leftarrow \mathbb{Z}_q^{n \times m} \]

\[ u = Ax \]

(public key, uniform when $m > n \log q$)
Public-Key Cryptosystem from LWE [Regev'05,GPV'08]

short \( x \)

\[
A \leftarrow \mathbb{Z}_q^{n \times m}
\]

\[
s \leftarrow \mathbb{Z}_q^n
\]

\[
u = Ax
\]

(public key, uniform when \( m > n \log q \))

\[
b^t = s^t A + e^t
\]

(ciphertext ‘preamble’)
Public-Key Cryptosystem from LWE \[\text{[Regev'05,GPV'08]}\]

\[
\text{short } x
\]

\[
A \leftarrow \mathbb{Z}_q^{n \times m}
\]

\[
s \leftarrow \mathbb{Z}_q^n
\]

\[
u = Ax
\]

(public key, uniform when \(m > n \log q\))

\[
b^t = s^t A + e^t
\]

(ciphertext 'preamble')

\[
b' = s^t u + e' + \text{bit} \cdot \frac{q}{2}
\]

('payload')
Public-Key Cryptosystem from LWE \[\text{[Regev'05,GPV'08]}\]

\[\begin{array}{c}
\text{short } x \\
A \leftarrow \mathbb{Z}_q^{n \times m} \\
s \leftarrow \mathbb{Z}_q^n \\
\end{array}\]

\[u = Ax\]
(public key, uniform when \(m > n \log q\))

\[b^t = s^t A + e^t\]
(ciphertext 'preamble')

\[b' - b^t x \approx \text{bit } \cdot \frac{q}{2}\]

\[b' = s^t u + e' + \text{bit } \cdot \frac{q}{2}\]
('payload')
Public-Key Cryptosystem from LWE  [Regev'05,GPV'08]

short $x$

$A \leftarrow \mathbb{Z}_{q}^{n \times m}$

$s \leftarrow \mathbb{Z}_{q}^{n}$

$u = Ax$

(public key, uniform when $m > n \log q$)

$b^{t} = s^{t}A + e^{t}$

(ciphertext ‘preamble’)

$b' = s^{t}u + e' + \text{bit} \cdot \frac{q}{2}$

('payload')

$b' - b^{t}x \approx \text{bit} \cdot \frac{q}{2}$

$(A, u), (b, b')$
Public-Key Cryptosystem from LWE \[\text{[Regev'05,GPV'08]}\]

\[\begin{align*}
A & \leftarrow \mathbb{Z}_q^{n \times m} \\
\mathbf{s} & \leftarrow \mathbb{Z}_q^n
\end{align*}\]

\[\mathbf{u} = A\mathbf{x}\] (public key, uniform when \(m > n \log q\))

\[\mathbf{b}^t = \mathbf{s}^t A + \mathbf{e}^t\] (ciphertext ‘preamble’)

\[b' = \mathbf{s}^t \mathbf{u} + e' + \text{bit} \cdot \frac{q}{2}\] (‘payload’)

\[(A, \mathbf{u}), (\mathbf{b}, b')\]

by LWE
Identity-Based Encryption

- Proposed by [Shamir’84]: could this exist?

\[ \text{mpk} \ (\text{msk}) \]
Identity-Based Encryption

- Proposed by [Shamir’84]: could this exist?

```
mpk (msk)
```

- [BonehFranklin’01, ...]
  - first IBE, based on pairings
- [Cocks’01, BGH’07]
  - based on quadratic residuosity mod $N = pq$
- [GPV’08]
  - based on lattices!
Identity-Based Encryption

- Proposed by [Shamir’84]: could this exist?

\[ \text{mpk} \ (\text{msk}) \]

\[ \text{Enc(mpk, “Alice”, msg)} \]

\[ s_k\text{Alice} \]

\[ s_k\text{Bobbi} \]

\[ s_k\text{Carol} \]
Identity-Based Encryption

- Proposed by [Shamir’84]: could this exist?

\[ \text{mpk (msk)} \]

\[ \text{sk}_{\text{Alice}} \quad \text{sk}_{\text{Bobbi}} \quad \text{sk}_{\text{Carol}} \]

\[ \text{Enc(mpk, “Alice”, msg)} \]
Identity-Based Encryption

- Proposed by [Shamir’84]: could this exist?

Enc(mpk, “Alice”, msg)

[1] [BonehFranklin’01,…]: first IBE, based on pairings
Identity-Based Encryption

- Proposed by [Shamir'84]: could this exist?

\[
\text{mpk (msk)}
\]

\[
\text{Enc(mpk, "Alice", msg)}
\]

1. [BonehFranklin'01, ...]: first IBE, based on pairings
2. [Cocks'01,BGH'07]: based on quadratic residuosity mod \( N = pq \)
Identity-Based Encryption

- Proposed by [Shamir’84]: could this exist?

\[
\text{Enc}(\text{mpk}, \text{Alice}, \text{msg})
\]

1. [BonehFranklin’01,…]: first IBE, based on pairings
2. [Cocks’01,BGH’07]: based on quadratic residuosity mod \( N = pq \)
3. [GPV’08]: based on lattices!
IBE from LWE

\[ \text{Gaussian } x \text{ s.t. } Ax = u \]

\[ mpk = A \]
\[ msk = \text{trapdoor } T \]

\[ u = H(“Alice”) \]
(‘identity’ public key)

\[ b = s^t A + e^t \]
(ciphertext preamble)

\[ b' - b^t x \approx \text{bit} \cdot \frac{q}{2} \]

\[ b' = s^t u + e' + \text{bit} \cdot \frac{q}{2} \]
(‘payload’)
Part 3:

Rings for Better Efficiency
SIS/LWE are (Sort Of) Efficient

\[
\begin{pmatrix}
\cdots & a_i & \cdots
\end{pmatrix}
\begin{pmatrix}
\vdots \\
s \\
\vdots
\end{pmatrix}
+ e_i = b_i \in \mathbb{Z}_q
\]

- Getting one pseudorandom scalar \( b_i \in \mathbb{Z}_q \) requires an \( n \)-dim inner product (mod \( q \))

Cryptosystems have rather large keys:

\[
\begin{pmatrix}
\cdots & A_t & \cdots \\
\cdots & b & \cdots
\end{pmatrix}
\]

\[
\mathcal{O}(n)
\]

Inherently \( \geq n^2 \) time to encrypt & decrypt a message.
SIS/LWE are (Sort Of) Efficient

\[
(\cdots a_i \cdots) \begin{pmatrix} \vdots \\ s \\ \vdots \end{pmatrix} + e_i = b_i \in \mathbb{Z}_q
\]

- Getting one pseudorandom scalar \( b_i \in \mathbb{Z}_q \) requires an \( n \)-dim inner product (mod \( q \))
- Can amortize each \( a_i \) over many secrets \( s_j \), but still \( \tilde{O}(n) \) work per scalar output.

Cryptosystems have rather large keys: 

\[
\begin{pmatrix} \vdots \\ A_t \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ b \end{pmatrix}
\] 

\[\Omega(n)\]

Inherently \( \geq n^2 \) time to encrypt & decrypt a message.
SIS/LWE are (Sort Of) Efficient

\[
\begin{pmatrix}
\cdots a_i \cdots \\
\vdots \\
\cdots 
\end{pmatrix}
\begin{pmatrix}
\vdots
\end{pmatrix}
+ e_i = b_i \in \mathbb{Z}_q
\]

- Getting one pseudorandom scalar \( b_i \in \mathbb{Z}_q \) requires an \( n \)-dim inner product (mod \( q \))
- Can amortize each \( a_i \) over many secrets \( s_j \), but still \( \tilde{O}(n) \) work per scalar output.
- Cryptosystems have rather large keys:

\[
pk = \begin{pmatrix}
\vdots \\
A^t \\
\vdots 
\end{pmatrix}, \quad \begin{pmatrix}
\vdots
\end{pmatrix} \Omega(n)
\]

Inherently \( \geq n^2 \) time to encrypt & decrypt a message.
SIS/LWE are (Sort Of) Efficient

\[
\begin{pmatrix}
\cdots & a_i & \cdots \\
\vdots & \ddots & \ddots \\
\end{pmatrix}
\begin{pmatrix}
\vdots \\
s \\
\vdots \\
\end{pmatrix}
+ e_i = b_i \in \mathbb{Z}_q
\]

- Getting one pseudorandom scalar \( b_i \in \mathbb{Z}_q \) requires an \( n \)-dim inner product (mod \( q \))
- Can amortize each \( a_i \) over many secrets \( s_j \), but still \( \tilde{O}(n) \) work per scalar output.

- Cryptosystems have rather large keys:

\[
pk = \begin{pmatrix}
\vdots \\
A^t \\
\vdots \\
\end{pmatrix}, \quad \begin{pmatrix}
\vdots \\
b \\
\vdots \\
\end{pmatrix}
\]

\( \Omega(n) \)

- Inherently \( \geq n^2 \) time to encrypt & decrypt a message.
Wishful Thinking...

\[
\begin{pmatrix}
\vdots \\
a_i \\
\vdots
\end{pmatrix} \ast \begin{pmatrix}
\vdots \\
s \\
\vdots
\end{pmatrix} + \begin{pmatrix}
\vdots \\
e_i \\
\vdots
\end{pmatrix} = \begin{pmatrix}
\vdots \\
b_i \\
\vdots
\end{pmatrix} \in \mathbb{Z}_q^n
\]

- Get \( n \) pseudorandom scalars from just one (cheap) product operation?
- Replace \( n \times n \) blocks by \( n \)-dimensional vectors.

▶ Careful! With small error, coordinate-wise multiplication is insecure!

Answer ▶ \( \ast \) = multiplication in a polynomial ring: e.g., \( \mathbb{Z}_q[X]/(X^n + 1) \).

Fast and practical with FFT: \( n \log n \) operations mod \( q \).

▶ Same ring structures used in NTRU cryptosystem [HPS'98], compact one-way / CR hash functions [Mic'02, PR'06, LM'06, ...]
Wishful Thinking... 

Get $n$ pseudorandom scalars from just one (cheap) product operation?

Replace $n \times n$ blocks by $n$-dimensional vectors.

Question

How to define the product ‘$\star$’ so that $(a_i, b_i)$ is pseudorandom?
Get $n$ pseudorandom scalars from just one (cheap) product operation?

Replace $n \times n$ blocks by $n$-dimensional vectors.

Question

- How to define the product ‘$\star$’ so that $(\mathbf{a}_i, \mathbf{b}_i)$ is pseudorandom?
- Careful! With small error, coordinate-wise multiplication is insecure!
Wishful Thinking... 

\[
\begin{pmatrix}
\vdots \\
a_i \\
\vdots
\end{pmatrix} \star 
\begin{pmatrix}
\vdots \\
s \\
\vdots
\end{pmatrix} + 
\begin{pmatrix}
\vdots \\
e_i \\
\vdots
\end{pmatrix} = 
\begin{pmatrix}
\vdots \\
b_i \\
\vdots
\end{pmatrix} \in \mathbb{Z}_q^n
\]

- Get \( n \) pseudorandom scalars from just one (cheap) product operation?
- Replace \( n \times n \) blocks by \( n \)-dimensional vectors.

Question

- How to define the product ‘\( \star \)’ so that \((a_i, b_i)\) is pseudorandom?
- Careful! With small error, coordinate-wise multiplication is insecure!

Answer

- ‘\( \star \)’ = multiplication in a polynomial ring: e.g., \( \mathbb{Z}_q[X]/(X^n + 1) \).
  Fast and practical with FFT: \( n \log n \) operations mod \( q \).
Wishful Thinking... 

\[
\begin{pmatrix}
\vdots \\
a_i \\
\vdots 
\end{pmatrix} \star \begin{pmatrix}
\vdots \\
s \\
\vdots 
\end{pmatrix} + \begin{pmatrix}
\vdots \\
e_i \\
\vdots 
\end{pmatrix} = \begin{pmatrix}
\vdots \\
b_i \\
\vdots 
\end{pmatrix} \in \mathbb{Z}_q^n
\]

- Get \( n \) pseudorandom scalars from just one (cheap) product operation?
- Replace \( n \times n \) blocks by \( n \)-dimensional vectors.

**Question**

- How to define the product ‘\( \star \)’ so that \((a_i, b_i)\) is pseudorandom?
- Careful! With small error, coordinate-wise multiplication is insecure!

**Answer**

- ‘\( \star \)’ = multiplication in a polynomial ring: e.g., \( \mathbb{Z}_q[X]/(X^n + 1) \).
  
  Fast and practical with FFT: \( n \log n \) operations mod \( q \).

- Same ring structures used in NTRU cryptosystem [HPS’98], compact one-way / CR hash functions [Mic’02,PR’06,LM’06,...]
LWE Over Rings, Over Simplified

Let $R = \mathbb{Z}[X]/(X^n + 1)$ for $n$ a power of two, and $R_q = R/qR$.
LWE Over Rings, Over Simplified

Let $R = \mathbb{Z}[X]/(X^n + 1)$ for $n$ a power of two, and $R_q = R/qR$

- Elements of $R_q$ are $\text{deg} < n$ polynomials with $\text{mod-}q$ coefficients
- Operations in $R_q$ are very efficient using FFT-like algorithms
LWE Over Rings, Over Simplified

Let \( R = \mathbb{Z}[X]/(X^n + 1) \) for \( n \) a power of two, and \( R_q = R/qR \)

- Elements of \( R_q \) are deg \( < n \) polynomials with mod-\( q \) coefficients
- Operations in \( R_q \) are very efficient using FFT-like algorithms

**Search:** find secret ring element \( s \in R_q \), given:

\[
\begin{align*}
    a_1 & \leftarrow R_q, & b_1 = s \cdot a_1 + e_1 & \in R_q \\
    a_2 & \leftarrow R_q, & b_2 = s \cdot a_2 + e_2 & \in R_q \\
    a_3 & \leftarrow R_q, & b_3 = s \cdot a_3 + e_3 & \in R_q \\
    \vdots
\end{align*}
\]

\((e_i \in R \text{ are ‘small’})\)
LWE Over Rings, Over Simplified

- Let $R = \mathbb{Z}[X]/(X^n + 1)$ for $n$ a power of two, and $R_q = R/qR$
  - Elements of $R_q$ are deg $< n$ polynomials with mod-$q$ coefficients
  - Operations in $R_q$ are very efficient using FFT-like algorithms

- **Search**: find secret ring element $s \in R_q$, given:

  $a_1 \leftarrow R_q$, $b_1 = s \cdot a_1 + e_1 \in R_q$
  $a_2 \leftarrow R_q$, $b_2 = s \cdot a_2 + e_2 \in R_q$
  $a_3 \leftarrow R_q$, $b_3 = s \cdot a_3 + e_3 \in R_q$

  $\vdots$

  $(e_i \in R$ are ‘small’)$

- **Decision**: distinguish $(a_i, b_i)$ from uniform $(a_i, b_i) \in R_q \times R_q$
Hardness of Ring-LWE

Initial Reductions [LyubashevskyPeikertRegev’10]

\[
\text{worst-case approx-SVP on } \text{ideal lattices in } R \leq \text{search } R\text{-LWE} \leq \text{decision } R\text{-LWE}
\]

(quantum, any \( R = \mathcal{O}_K \))

(classical, any cyclotomic \( R \))
Hardness of Ring-LWE

**Initial Reductions** [LyubashevskyPeikertRegev’10]

\[
\text{worst-case approx-SVP on ideal lattices in } R \leq \text{search } R\text{-LWE} \leq \text{decision } R\text{-LWE}
\]

(qquantum, any } R = \mathcal{O}_K) (classical, any cyclotomic } R)

**Newer Reduction** [PeikertRegevStephens-Davidowitz’17]

\[
\text{worst-case approx-SVP on ideal lattices in } R \leq \text{decision } R\text{-LWE}
\]

(qquantum, any } R = \mathcal{O}_K)
Hardness of Ring-LWE

Initial Reductions [LyubashevskyPeikertRegev’10]

\[
\text{worst-case } \text{approx-SVP on ideal lattices in } R \leq \text{ search } R\text{-LWE} \leq \text{ decision } R\text{-LWE} \\
\text{(quantum, any } R = \mathcal{O}_K) \quad \text{(classical, any cyclotomic } R)\]

Newer Reduction [PeikertRegevStephens-Davidowitz’17]

\[
\text{worst-case } \text{approx-SVP on ideal lattices in } R \leq \text{ decision } R\text{-LWE} \\
\text{(quantum, any } R = \mathcal{O}_K)\]

Constructions

\[
\text{decision } R\text{-LWE} \leq \text{ much crypto}\]
Final Thoughts

- Lattices are a **very attractive foundation** for post-quantum crypto, for both ‘basic’ and ‘advanced’ objects.

See remaining talks for much more.
Final Thoughts

- Lattices are a very attractive foundation for post-quantum crypto, for both ‘basic’ and ‘advanced’ objects.
  
  See remaining talks for much more.

- Cryptanalysis\textit{concrete security estimates} are subtle and ongoing, but maturing.
  
  See Phong Nguyen’s talks tomorrow for coverage of this topic.
Final Thoughts

- Lattices are a very attractive foundation for post-quantum crypto, for both ‘basic’ and ‘advanced’ objects. See remaining talks for much more.

- Cryptanalysis/concrete security estimates are subtle and ongoing, but maturing. See Phong Nguyen’s talks tomorrow for coverage of this topic.

Thanks!