Ideal Lattices

Vadim Lyubashevsky
Tel-Aviv University
Ideal Lattice FAQs

Q: What are ideal lattices?
A: They are lattices with some additional algebraic structure.
  Lattices are groups
  Ideal Lattices are ideals

Q: What can we do with ideal lattices?
A: 1. Build efficient cryptographic primitives
   2. Build a homomorphic encryption scheme
Cyclic Lattices

A set \( L \) in \( \mathbb{Z}^n \) is a *cyclic lattice* if:

1.) For all \( v, w \) in \( L \), \( v+w \) is also in \( L \)

\[
\begin{array}{cccc}
-1 & 2 & 3 & -4 \\
\end{array}
\quad +
\begin{array}{cccc}
-7 & -2 & 3 & 6 \\
\end{array}
=\begin{array}{cccc}
-8 & 0 & 6 & 2 \\
\end{array}
\]

2.) For all \( v \) in \( L \), \(-v\) is also in \( L \)

\[
\begin{array}{cccc}
-1 & 2 & 3 & -4 \\
\end{array}
\quad \quad
\begin{array}{cccc}
1 & -2 & -3 & 4 \\
\end{array}
\]

3.) For all \( v \) in \( L \), a cyclic shift of \( v \) is also in \( L \)

\[
\begin{array}{cccc}
-1 & 2 & 3 & -4 \\
\hline
-4 & -1 & 2 & 3 \\
\hline
3 & -4 & -1 & 2 \\
\hline
2 & 3 & -4 & -1 \\
\end{array}
\]
Cyclic Lattices $= \text{Ideals in } \mathbb{Z}[x]/(x^n-1)$

A set $L$ in $\mathbb{Z}^n$ is a cyclic lattice if $L$ is an ideal in $\mathbb{Z}[x]/(x^n-1)$

1.) For all $v,w$ in $L$, $v+w$ is also in $L$

$$\begin{bmatrix} -1 & 2 & 3 & -4 \\ -7 & -2 & 3 & 6 \end{bmatrix} = \begin{bmatrix} -8 & 0 & 6 & 2 \end{bmatrix}$$

$$(-1+2x+3x^2-4x^3)+(-7-2x+3x^2+6x^3)=(-8+0x+6x^2+2x^3)$$

2.) For all $v$ in $L$, $-v$ is also in $L$

$$\begin{bmatrix} -1 & 2 & 3 & -4 \\ 1 & -2 & -3 & 4 \end{bmatrix}$$

$$(-1+2x+3x^2-4x^3) \quad (1-2x-3x^2+4x^3)$$

3.) For all $v$ in $L$, a cyclic shift of $v$ is also in $L$ $vx$ is also in $L$

$$\begin{bmatrix} -1 & 2 & 3 & -4 \\ -4 & -1 & 2 & 3 \\ 3 & -4 & -1 & 2 \\ 2 & 3 & -4 & -1 \end{bmatrix}$$

$$(-1+2x+3x^2-4x^3)x = -4x+2x^2+3x^3$$

$$(-1+2x+3x^2-4x^3)x^2 = 3-4x-x^2+2x^3$$

$$(-1+2x+3x^2-4x^3)x^3 = 2+3x-4x^2-x^3$$
Why Cyclic Lattices?

• Succinct representations
  – Can represent an n-dimensional lattice with 1 vector

• Algebraic structure
  – Allows for fast arithmetic (using FFT)
  – Makes proofs possible

• NTRU cryptosystem (fast but no proofs)

• One-way functions based on the worst-case hardness of SVP in cyclic lattices [Mic02]
Is $\text{SVP}_{\text{poly}(n)}$ Hard for Cyclic Lattices?

Short answer: we don't know but conjecture it is.

What's wrong with the following argument that $\text{SVP}_n$ is easy?

Algorithm for solving $\text{SVP}_n(L)$ for a cyclic lattice $L$:
1. Construct 1-dimensional lattice $L' = L \cap \{1^n\}$
2. Find and output the shortest vector in $L'$
The Hard Cyclic Lattice Instances

The “hard” instances of cyclic lattices lie on plane P perpendicular to the $1^n$ vector.

In algebra language:

If $R = \mathbb{Z}[x]/(x^n-1)$, then

$$1^n = (x^{n-1} + x^{n-2} + \ldots + 1) \approx R/(x-1) \approx \mathbb{Z}[x]/(x-1)$$

$$P = (x-1) \approx R/(x^{n-1} + x^{n-2} + \ldots + 1) \approx \mathbb{Z}[x]/(x^{n-1} + x^{n-2} + \ldots + 1)$$
f-Ideal Lattices = Ideals in $\mathbb{Z}[x]/(f)$

Want $f$ to have 3 properties:

1) Monic (i.e. coefficient of largest exponent is 1)
2) Irreducible over $\mathbb{Z}$
3) For all polynomials $g, h$ $||gh \mod f|| < \text{poly}(n)||g||^*||h||$

Conjecture: For all $f$ that satisfy the above 3 properties, solving SVP$_{\text{poly}(n)}$ for ideals in $\mathbb{Z}[x]/(f)$ takes time $2^{\Omega(n)}$.

Some “good” $f$ to use:

$f = x^{n-1} + x^{n-2} + \ldots + 1$ where $n$ is prime

$f = x^n + 1$ where $n$ is a power of 2
(x^n+1)-Ideal Lattices = Ideals in \( \mathbb{Z}[x]/(x^n+1) \)

A set \( L \) in \( \mathbb{Z}^n \) is an \((x^n+1)\)-ideal lattice if \( L \) is an ideal in \( \mathbb{Z}[x]/(x^n+1) \)

1.) For all \( v, w \) in \( L \), \( v+w \) is also in \( L \)

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\end{array}
\quad + \quad
\begin{array}{cccc}
-7 & -2 & 3 & 6 \\
\end{array}
\quad = \quad
\begin{array}{cccc}
-6 & 0 & 6 & 10 \\
\end{array}
\]

\((1+2x+3x^2+4x^3)+(-7-2x+3x^2+6x^3)=(-6+0x+6x^2+10x^3)\)

2.) For all \( v \) in \( L \), \(-v\) is also in \( L \)

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\end{array}
\quad - \quad
\begin{array}{cccc}
-1 & -2 & -3 & -4 \\
\end{array}
\]

\((1+2x+3x^2+4x^3) \quad (-1-2x-3x^2-4x^3)\)

3.) For all \( v \) in \( L \), \( vx \) is also in \( L \)

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\end{array}
\quad \times \quad
\begin{array}{cccc}
1+2x+3x^2+4x^3 \\
\end{array}
\]

\((1+2x+3x^2+4x^3)x= -4+x+2x^2+3x^3\)

\[
\begin{array}{cccc}
-4 & 1 & 2 & 3 \\
\end{array}
\quad \times \quad
\begin{array}{cccc}
(1+2x+3x^2+4x^3)x= -4+x+2x^2+3x^3 \\
\end{array}
\]

\((1+2x+3x^2+4x^3)x^2 = -3-4x+x^2+2x^3\)

\[
\begin{array}{cccc}
-3 & -4 & 1 & 2 \\
\end{array}
\quad \times \quad
\begin{array}{cccc}
(1+2x+3x^2+4x^3)x^2 = -3-4x+x^2+2x^3 \\
\end{array}
\]

\((1+2x+3x^2+4x^3)x^3 = -2-3x-4x^2+x^3\)

\[
\begin{array}{cccc}
-2 & -3 & -4 & 1 \\
\end{array}
\quad \times \quad
\begin{array}{cccc}
(1+2x+3x^2+4x^3)x^3 = -2-3x-4x^2+x^3 \\
\end{array}
\]
# Hardness of Problems for General and \((x^n+1)\)-Ideal Lattices

## Exact Versions

<table>
<thead>
<tr>
<th>Problem</th>
<th>General</th>
<th>((x^n+1))-ideal</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVP</td>
<td>NP-hard</td>
<td>?</td>
</tr>
<tr>
<td>SIVP</td>
<td>NP-hard</td>
<td>?</td>
</tr>
<tr>
<td>GapSVP</td>
<td>NP-hard</td>
<td>?</td>
</tr>
<tr>
<td>uSVP</td>
<td>NP-hard</td>
<td>N/A</td>
</tr>
<tr>
<td>BDD</td>
<td>NP-hard</td>
<td>?</td>
</tr>
</tbody>
</table>

## Poly(n)-approximate Versions

<table>
<thead>
<tr>
<th>Problem</th>
<th>General</th>
<th>((x^n+1))-ideal</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVP</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>SIVP</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>GapSVP</td>
<td>?</td>
<td>Easy</td>
</tr>
<tr>
<td>uSVP</td>
<td>?</td>
<td>N/A</td>
</tr>
<tr>
<td>BDD</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

**Legend:**

?: No hardness proofs nor sub-exponential time algorithms are known.
Colored boxes: Problems are equivalent
Lemma: If $v$ is a vector in $\mathbb{Z}[x]/(f)$ where $f$ is a monic, irreducible polynomial of degree $n$, then $v, vx, vx^2, \ldots vx^{n-1}$ are linearly independent.

Proof: Suppose not. Let $v$ be in $\mathbb{Z}[x]$ with $\deg(v) < n$, and $a_0, a_1, a_2, \ldots a_{n-1}$ in $\mathbb{Z}$ such that

$$a_0v + a_1vx + a_2vx^2 + \ldots + a_{n-1}vx^{n-1} \mod f = 0$$

$$v(a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1}) \mod f = 0$$

$$vw \mod f = 0$$

$f$ is irreducible (also prime), thus either $f|v$ or $f|w$. But $\deg(v), \deg(w) < n$, so contradiction.
SVP = SIVP

Lemma: If \( v \) is a vector in \( \mathbb{Z}[x]/(f) \) where \( f \) is a monic, irreducible polynomial of degree \( n \), then

\[ v, vx, vx^2, \ldots, vx^{n-1} \]

are linearly independent.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>-3</td>
<td>-4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>1</td>
</tr>
</tbody>
</table>

Shortest vector \( v \)

\[ ||v|| = ||vx|| = ||vx^2|| = ||vx^3|| \]

Corollary: A \( (x^n+1) \)-ideal lattice cannot have a unique shortest vector.
GapSVP$_{\sqrt{n}}$ is easy

Fact: For all $(x^n+1)$-ideal lattices $L$,
\[ \det(L)^{1/n} \leq \lambda_1(L) \leq \sqrt{n} \det(L)^{1/n} \]

So $\det(L)^{1/n}$ is a $\sqrt{n}$ – approximation of $\lambda_1(L)$

Proof of fact:
1. $\lambda_1(L) \leq \sqrt{n} \det(L)^{1/n}$ is Minkowski's theorem.
2. Let $v$ be the shortest vector of $L$. Define $L'=(v)$.
   (i.e. $L'$ is generated by vectors $v$, $vx$, $vx^2$, ... $vx^{n-1}$)

$L'$ is a sublattice of $L$, so we have
\[ \det(L) \leq \det(L') \leq ||v||^n = \left( \lambda_1(L) \right)^n \]
Applications of Ideal Lattices

- One-way functions based on SVP [Mic02]
- Collision-resistant hash functions based on SVP [LM06, PR06, LMPR08, ADLMPR08]
- Tighter worst-case to average-case reductions [PR07]
- One-time signatures based on SVP [LM08]
- Almost practical ID and signature schemes based on SVP [Lyu08]
- Fully homomorphic encryption based on BDD [Gen09]
- Encryption schemes based on quantum hardness of SVP [SSTX09]
Collision-Resistant Hash Function

- Collision-resistant hash function [LM06, PR06, LMPR08]
  - Provable security based on worst-case hardness of approximating $\text{SVP}_{\tilde{O}(n)}$
  - Function evaluation in $\tilde{O}(n)$ time vs. $\tilde{O}(n^2)$ for general lattices
  - SWIFFTX hash function entered into SHA-3 competition. Efficient in practice. [ADLMPR08]
Choose \( p \) to be a number \( \approx O(n^{1.5}) \)

Choose elements \( a_1, \ldots, a_{3\log(n)} \) randomly in \( \mathbb{Z}_p[x]/(x^n+1) \)

On an input from \( \{0,1\}^{3n\log(n)} \):

\[
\begin{array}{cccc}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
\end{array} \quad \cdots \quad \begin{array}{cccc}
1 & 1 & 0 & 0 \\
\end{array}
\]

\[
y_1 \quad y_2 \quad \cdots \quad y_{3\log(n)}
\]

Output: \( a_1 y_1 + a_2 y_2 + \ldots + a_{3\log(n)} y_{3\log(n)} \mod p \)

Function maps \( 3n\log(n) \) bits to \( \log(p^n) = n\log(p) = 1.5n\log(n) \) bits
Efficiency of the Hash Function

- The hash function is defined by $O(\log(n))$ elements in $\mathbb{Z}_p[x]/(x^n+1)$
  - Each element requires $n \log(p)$ bits
  - Total space needed $O(n \log^2 n)$ bits
- Computing $a_1 y_1 + a_2 y_2 + ... + a_{3\log(n)} y_{3\log(n)}$ requires
  - $3 \log(n)$ additions: $O(n \log^2 n)$ time
  - $3 \log(n)$ multiplications: $O(n \log^3 n)$ time using FFT
- In practice
  - Can exploit parallelism
  - Can do a lot of pre-processing for the FFT
# Comparison of Lattice Hash Functions

<table>
<thead>
<tr>
<th></th>
<th>General Lattices ([Ajt96, ..., MR07])</th>
<th>((x^n+1))-ideal lattices ([LM06, PR06, LMPR08])</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage</strong></td>
<td>(\tilde{O}(n^2))</td>
<td>(\tilde{O}(n))</td>
</tr>
<tr>
<td><strong>Computing Time</strong></td>
<td>(\tilde{O}(n^2))</td>
<td>(\tilde{O}(n))</td>
</tr>
<tr>
<td><strong>Hardness Assumption</strong></td>
<td>SIVP(\tilde{o}(n)) or GapSVP(\tilde{o}(n))</td>
<td>((x^n+1))-ideal SVP(\tilde{o}(n))</td>
</tr>
<tr>
<td><strong>Best Known Attack Time</strong></td>
<td>(2^{\Omega(n)})</td>
<td>(2^{\Omega(n)})</td>
</tr>
</tbody>
</table>
Proof of Security

- Finding collisions in a random hash function instance $a_1,...,a_{3\log(n)}$ (for $a_i$ in $\mathbb{Z}_p[x]/(x^n+1)$) is as hard as solving $\text{SVP}_{\tilde{O}(n)}$ in any ideal of $\mathbb{Z}[x]/(x^n+1)$
- Proof similar to the one for general lattices
- Proceed in iterations:
  1) Have some vector in $L$
  2) Create a random hash function
  3) Finding a collision $\rightarrow$ finding a shorter vector
  4) Repeat
Security Proof
(From one vector to n vectors)

\[ \|v\| > n \lambda_1(L) \]

The vectors \( v, vx, vx^2, \ldots vx^{n-1} \) generate a full-dimensional sub-lattice of \( L \)

For simplicity, we'll assume that they generate \( L \)
Repeat 3\log(n) times:

1. Generate a point close to the origin according to a Gaussian distribution with a “large-enough” variance
2. Reduce the point into the parallelepiped
   (By [MR07], the points are statistically close to uniform in \( \mathbb{R}^n/L \))
Security Proof

(Getting a random hash function)

Result of sampling:

Have randomly-distributed points in the parallelepiped
Each element has a lattice point “not too far away”
(approximately $\|v\|/\sqrt{n} > \sqrt{n} \lambda_1(L)$ away)
Subdivide each side of the parallelepiped into $p$ divisions. Each intersection corresponds to an element in $\mathbb{Z}_p[x]/(x^n+1)$.

Round each generated point to the nearest intersection. The $3\log(n)$ intersection points define the random hash function.
Security Proof
(Finding a collision → finding a shorter vector)

Have a random hash function defined by
\[ a_1, ..., a_{3\log(n)} \]

Suppose we find a collision
\[ a_1 y_1 + ... + a_{3\log(n)} y_{3\log(n)} = a_1 y'_1 + ... + a_{3\log(n)} y'_{3\log(n)} \mod p \]

where \( y_i, y'_i \) have 0/1 coefficients

Then
\[ a_1 (y_1 - y'_1) + ... + a_{3\log(n)} (y_{3\log(n)} - y'_{3\log(n)}) = 0 \mod p \]

So,
\[ a_1 z_1 + ... + a_{3\log(n)} z_{3\log(n)} = 0 \mod p \]

where \( z_i \) have -1/0/1 coefficients
Security Proof
(Finding a collision → finding a shorter vector)

Consider $h = w_1z_1 + \ldots + w_{3\log(n)}z_{3\log(n)}$ (h is in L because $w_i$ are in L and $z_i$ are in $\mathbb{Z}[x]/(x^n+1)$).

$h = (r_1 + (v_{a_1}/p))z_1 + \ldots + (r_{3\log(n)} + (v_{a_{3\log(n)}})/p)z_{3\log(n)}$

$= r_1z_1 + \ldots r_{3\log(n)}z_{3\log(n)} + v(a_1z_1 + \ldots + a_{3\log(n)}z_{3\log(n)})/p$

$= r_1z_1 + \ldots r_{3\log(n)}z_{3\log(n)} + vpg/p$ for some $g$ in $\mathbb{Z}[x]/(x^n+1)$ (for $v_{a_i}/p$ (multiplication over $\mathbb{R}[x]/(x^n+1)$)

$= r_1z_1 + \ldots r_{3\log(n)}z_{3\log(n)} + vg$

So $r_1z_1 + \ldots r_{3\log(n)}z_{3\log(n)}$ is in L

Lattice point $w_i$ close to $(v_{a_i}/p)$

$r_i = w_i - (v_{a_i}/p)$
Security Proof
(Finding a collision → finding a shorter vector)

Found a vector \( r_1 z_1 + ... r_{3\log(n)} z_{3\log(n)} \)

How big is it?
\( z_i \) have \(-1/0/1\) coefficients (that's small)

How big are \( r_i \)?

So \( ||r_i|| \) is on the order of \( ||v||/\sqrt{n} \)
Security Proof
(Finding a collision $\rightarrow$ finding a shorter vector)

Using the fact that $r_i$ are chosen randomly, and the fact that $\|r_i\|$ is on the order of $\|v\|/\sqrt{n}$,

$$\|r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)}\| = O(\|v\|)$$

By modifying a few variables by polylog terms, we can make it strictly less than $\|v\|$

One more thing... need to make sure it's not 0

(Same idea as for general lattices)
One-time Signatures

- Nearly-optimal (asymptotically) 1-time signatures [LM08]
  - Signing and verification takes $\tilde{O}(n)$ time.
  - Breaking signature is conjectured to be $2^{\Omega(n)}$-hard
  - No other such constructions (even ad-hoc) are known
  - A black box conversion from 1-way functions would require $\Omega(n^2)$ time for $2^{\Omega(n)}$-security [BM08]

- Our construction:
  - Based on the hardness of finding collisions in the ideal lattice based hash function
  - Similar in spirit to some number-theoretic constructions
Modules and Hash Functions

Module: Like a *vector space*, but scalars can be in a ring instead of a field

Module $M=(G,R)$

$G$ is an Abelian group. $R$ is a ring.

Module homomorphism $h: M_1 \rightarrow M_2$ satisfies:

- $h(gr) = h(g)r$
- $h(g_1 + g_2) = h(g_1) + h(g_2)$

Hardness assumption: hard to find $g_1, g_2$ such that $h(g_1) = h(g_2)$
One-time Signature Scheme

\[ M_1 = (G_1, R_1) \]

Generate \( g_1, g_2 \) randomly in \( G_1 \)

Secret Key = \((g_1, g_2)\)

Public Key = \((h(g_1), h(g_2))\)

Message \( r \) in \( R_1 \)

Signature of \( r \) is \( s = g_1r + g_2 \)

Accept if \( h(s) = h(g_1)r + h(g_2) \)

Security proof idea:

Suppose an adversary finds message \( r' \) and signature \( s' \)

Then we can sign \( r' \) and the hash of our signature should equal to \( h(r') \)
What if \( s' = g_1 r' + g_2 \) ?

\[
M_1 = (G_1, R_1)
\]

Attacker knows:

- \( s = g_1 r + g_2 \)
- \( s' = g_1 r' + g_2 \)

\[
g_1 = \frac{(s - s')}{(r - r')}
\]

\[
g_2 = s - g_1 r
\]

Not giving us a collision implies knowing \( g_1 \) and \( g_2 \)
$g_1$ and $g_2$ are information-theoretically hidden

$M_1 = (G_1, R_1)$

Attacker knows:
- $r$
- $h(g_1)$
- $h(g_2)$
- $s = g_1r + g_2$

Let $z$ be in the kernel of $h$ (i.e. $h(z) = 0$)

Consider:
- $g_1' = g_1 - z$
- $g_2' = g_2 + zr$
Can we do this for ideal lattices?

- \( M = (G, R) \)
  - \( R = \mathbb{Z}_p[x]/(x^n + 1) \)
  - \( G = R^{3\log(n)} \)

- \( h(y_1, ..., y_{3\log(n)}) = a_1 y_1 + ... + a_{3\log(n)} y_{3\log(n)} \mod p \)

- Is \( h \) collision-resistant?
  - No. It's easy to find \((y_1, ..., y_{3\log(n)})\) and \((y'_1, ..., y'_{3\log(n)})\) such that \( h(y_1, ..., y_{3\log(n)}) = h(y'_1, ..., y'_{3\log(n)}) \)
  - It's hard to find small \((y_1, ..., y_{3\log(n)})\) and \((y'_1, ..., y'_{3\log(n)})\) such that \( h(y_1, ..., y_{3\log(n)}) = h(y'_1, ..., y'_{3\log(n)}) \)
One-time Signature Scheme

\[ M_1 = (G_1, R_1) \]

Generate short \( g_1, g_2 \) randomly in \( G_1 \)

Secret Key = \( (g_1, g_2) \)

Public Key = \( (h(g_1), h(g_2)) \)

Message is a short \( r \) in \( R_1 \)

Signature of \( r \) is \( s = g_1r + g_2 \)

Accept if \( h(s) = h(g_1)r + h(g_2) \) and \( s \) is small

Security proof idea:

Suppose an adversary finds message \( r' \) and signature \( s' \)

Then we can sign \( r' \) and the hash of our signature should equal to \( h(r') \)
$g_1$ and $g_2$ are information-theoretically hidden

$M_1 = (G_1, R_1)$

Attacker knows:

- $r$
- $h(g_1)$
- $h(g_2)$
- $s = g_1 r + g_2$

Let $z$ be in the kernel of $h$ (i.e. $h(z) = 0$)

Consider:

- $g_1' = g_1 - z$
- $g_2' = g_2 + z r$

Issue: $g_1'$, $g_2'$ may not be valid secret keys!
Making the lattice scheme work

• Intuitively,
  – Choose secret keys using a distribution such that larger keys are always possible
  – Expected key size is small
  – For any public key and signature, no secret key has too high a prior probability
Some Open Problems

- Design truly practical schemes based on ideal lattices
  - May involve making additional assumptions
- Prove some hardness results for ideal lattice problems
  - If that fails, make up a problem that's hard for ideal lattices
- Prove some non-hardness results for ideal lattice problems
  - e.g. show that $\text{SVP}_k$ is not NP hard for $k<\sqrt{n}$
- Show that solving SVP in ideals of $\mathbb{Z}[x]/(f)$ is easy for certain $f$
  - Might be a good idea to look at $f$ that are “very reducible”
- Does quantum computing help?
  - Ideal lattices have a lot more structure than general lattices
- Design more cryptographic primitives based on ideal lattice problems
  - Almost everything can be done with general lattices. Very few things can be done with ideal lattices