Introduction to Lattice Based Cryptography

Dragoș Alin Rotaru

Bitdefender Romania, University of Bucharest

October 21, 2015

Outline

- What is a lattice?
- Lattices in practice.
- Examples of hard problems on lattices.
- (Known) Algorithms for solving hard problems on lattices.
- (Maybe) NTRU cryptosystem.

Motivation - Post-Quantum Crypto

source: SafeCrypto Project

Overview of lattice-based constructions

- Fast and Efficient but lack of security proofs (NTRU).

Overview of lattice-based constructions

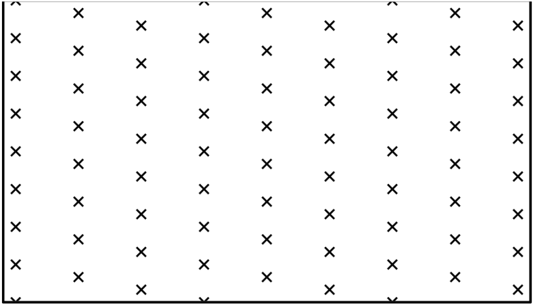
- Fast and Efficient but lack of security proofs (NTRU).
- Strong security proofs but not so fast (Learning with Errors).

Overview of lattice-based constructions

- Fast and Efficient but lack of security proofs (NTRU).
- Strong security proofs but not so fast (Learning with Errors).
- Searching for a solution from both worlds (Ring learning with Errors).

What is a lattice? v.1

Short Answer: A grid.



Lattice in R^2

What is a lattice? v.2

- The set of all linear integers combinations by some vectors in R^m.

What is a lattice? v.2

- The set of all linear integers combinations by some vectors in R^m.
- Given n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ the lattice generated by them is $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum x_i b_i, x_i \in \mathbb{Z}\}$.

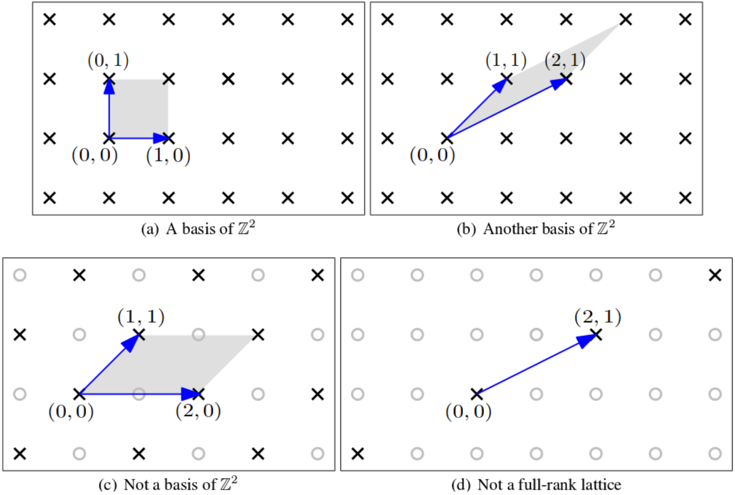
What is a lattice? v.2

- The set of all linear integers combinations by some vectors in R^m.
- Given n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ the lattice generated by them is $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum x_i b_i, x_i \in \mathbb{Z}\}$.
- We call $\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_n$ the basis of $\mathcal{L}$.

What is a lattice? v.2

- The set of all linear integers combinations by some vectors in R^m.
- Given n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ the lattice generated by them is $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum x_i b_i, x_i \in \mathbb{Z}\}$.
- We call $\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_n$ the basis of $\mathcal{L}$.
- Rewrite the definition as $\mathcal{L} = Bx$ where B has n columns: $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$.

Lattice Basis



(a) A basis of $\mathbb{Z}^2$

(b) Another basis of $\mathbb{Z}^2$

(c) Not a basis of $\mathbb{Z}^2$

(d) Not a full-rank lattice

Different bases - Source: Regev course

Lattice Basis

Fact

$\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists an unimodular integer matrix $U \in \mathbb{Z}^{nxn}$ such that $B = B'U$.

Lattice Basis

Fact

$\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists an unimodular integer matrix $U \in \mathbb{Z}^{nxn}$ such that $B = B'U$.

- Because U is unimodular, $det(U) = \pm 1$.

Lattice Basis

Fact

$\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists an unimodular integer matrix $U \in \mathbb{Z}^{nxn}$ such that $B = B'U$.

- Because U is unimodular, $det(U) = \pm 1$.
- So what?

Lattice Basis

Fact

$\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists an unimodular integer matrix $U \in \mathbb{Z}^{nxn}$ such that $B = B'U$.

- Because U is unimodular, $det(U) = \pm 1$.
- So what?
- $det(B)$ it's invariant over the choice of basis. Denote $det(\mathcal{L}) := |det(B)|$.

Lattice Basis

Fact

$\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists an unimodular integer matrix $U \in \mathbb{Z}^{nxn}$ such that $B = B'U$.

- Because U is unimodular, $det(U) = \pm 1$.
- So what?
- $det(B)$ it's invariant over the choice of basis. Denote $det(\mathcal{L}) := |det(B)|$.
- $det(\mathcal{L})$ is also called the fundamental volume of $\mathcal{L}$.

Lattice Basis

Fact

$\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists an unimodular integer matrix $U \in \mathbb{Z}^{nxn}$ such that $B = B'U$.

- Because U is unimodular, $det(U) = \pm 1$.
- So what?
- $det(B)$ it's invariant over the choice of basis. Denote $det(\mathcal{L}) := |det(B)|$.
- $det(\mathcal{L})$ is also called the fundamental volume of $\mathcal{L}$.
- Determinant of a lattice is inverse proportional to its density.

Succesive minima

Shortest Vector

$\lambda_1(\mathcal{L}) = min\{\|x\| : x \in \mathcal{L}, x \neq 0\}$

Succesive minima

Shortest Vector

$\lambda_1(\mathcal{L}) = min\{\|x\| : x \in \mathcal{L}, x \neq 0\}$

Succesive minima

$\lambda_i(\mathcal{L}) = min\{r : dim(span(\mathcal{L} \cap B(0, r))) \geq i\}$

Succesive minima

Shortest Vector

$\lambda_1(\mathcal{L}) = min\{\|x\| : x \in \mathcal{L}, x \neq 0\}$

Succesive minima

$\lambda_i(\mathcal{L}) = min\{r : dim(span(\mathcal{L} \cap B(0, r))) \geq i\}$

Upper bounds for $\lambda_1(\mathcal{L})$

For any lattice of dimension n, $\lambda_1(\mathcal{L}) \leq \sqrt{n}(det(\mathcal{L}))^{1/n}$.

Succesive minima

Shortest Vector

$\lambda_1(\mathcal{L}) = min\{\|x\| : x \in \mathcal{L}, x \neq 0\}$

Succesive minima

$\lambda_i(\mathcal{L}) = min\{r : dim(span(\mathcal{L} \cap B(0, r))) \geq i\}$

Upper bounds for $\lambda_1(\mathcal{L})$

For any lattice of dimension n, $\lambda_1(\mathcal{L}) \leq \sqrt{n}(det(\mathcal{L}))^{1/n}$.

- Unfortunately, no constructive proof.

Succesive minima

Shortest Vector

$\lambda_1(\mathcal{L}) = min\{\|x\| : x \in \mathcal{L}, x \neq 0\}$

Succesive minima

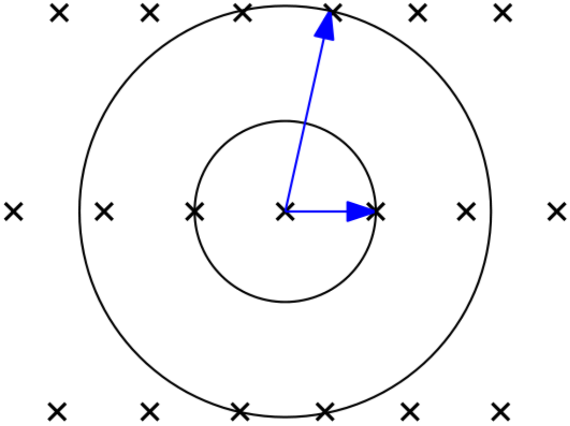
$\lambda_i(\mathcal{L}) = min\{r : dim(span(\mathcal{L} \cap B(0,r))) \geq i\}$

Upper bounds for $\lambda_1(\mathcal{L})$

For any lattice of dimension n, $\lambda_1(\mathcal{L}) \leq \sqrt{n}(det(\mathcal{L}))^{1/n}$.

- Unfortunately, no constructive proof.
- Also, a loose bound. Think about the lattice generated in $\mathbb{R}^2$ by
$$\begin{bmatrix} 0 & \epsilon \\ 1/\epsilon & 0 \end{bmatrix}$$

Succesive minima



$\lambda_1(\mathcal{L}), \lambda_2(\mathcal{L})$ - Source: Regev course

Dream world: Gram-Schmidt for Lattices

- It would be great to have a basis $\mathbf{b}_1 = \lambda_1(\mathcal{L}), \mathbf{b}_2 = \lambda_2(\mathcal{L}), \ldots, \mathbf{b}_n = \lambda_n(\mathcal{L})$.

Dream world: Gram-Schmidt for Lattices

- It would be great to have a basis $\mathbf{b}_1 = \lambda_1(\mathcal{L}), \mathbf{b}_2 = \lambda_2(\mathcal{L}), \ldots, \mathbf{b}_n = \lambda_n(\mathcal{L})$.
- Because the fundamental volume of $\mathcal{L}$ is invariant over the change of basis, short and orthogonal are related notions.

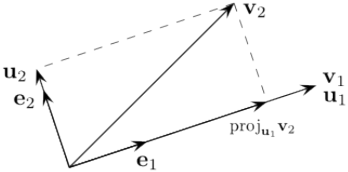
Dream world: Gram-Schmidt for Lattices

- It would be great to have a basis $\mathbf{b}_1 = \lambda_1(\mathcal{L}), \mathbf{b}_2 = \lambda_2(\mathcal{L}), \ldots, \mathbf{b}_n = \lambda_n(\mathcal{L})$.
- Because the fundamental volume of $\mathcal{L}$ is invariant over the change of basis, short and orthogonal are related notions.
- Solution: Let's apply Gram-Schmidt to a lattice basis!

Dream world: Gram-Schmidt for Lattices

- It would be great to have a basis $\mathbf{b}_1 = \lambda_1(\mathcal{L}), \mathbf{b}_2 = \lambda_2(\mathcal{L}), \ldots, \mathbf{b}_n = \lambda_n(\mathcal{L})$.
- Because the fundamental volume of $\mathcal{L}$ is invariant over the change of basis, short and orthogonal are related notions.
- Solution: Let's apply Gram-Schmidt to a lattice basis!
- What?

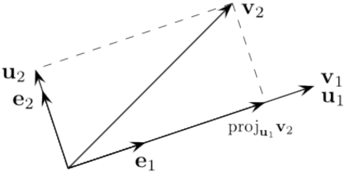
Gram-Schmidt Ortogonalization



Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$

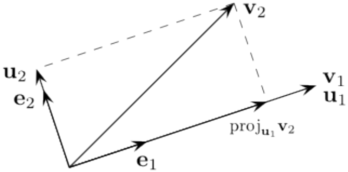
Gram-Schmidt Ortogonalization



Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$
- Output: $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$, such that $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0, i \neq j$

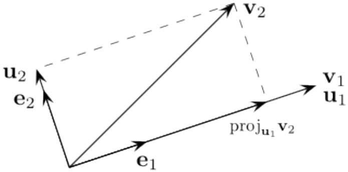
Gram-Schmidt Ortogonalization



Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$
- Output: $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$, such that $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0, i \neq j$
- Define the projection operator: $proj_u(v) = \frac{\langle v,u \rangle}{\langle u,u \rangle} u$.

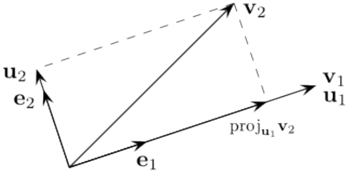
Gram-Schmidt Ortogonalization



Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$
- Output: $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$, such that $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0, i \neq j$
- Define the projection operator: $proj_u(v) = \frac{\langle v,u \rangle}{\langle u,u \rangle} u$.
- $\tilde{\mathbf{b}}_1 = \mathbf{b}_1, \tilde{\mathbf{b}}_2 = \mathbf{b}_2 - proj_{\tilde{\mathbf{b}}_1}(\mathbf{b}_2)$ and so on.

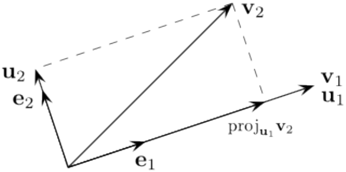
Gram-Schmidt Ortogonalization



Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$
- Output: $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$, such that $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0, i \neq j$
- Define the projection operator: $proj_u(v) = \frac{\langle v,u \rangle}{\langle u,u \rangle} u$.
- $\tilde{\mathbf{b}}_1 = \mathbf{b}_1, \tilde{\mathbf{b}}_2 = \mathbf{b}_2 - proj_{\tilde{\mathbf{b}}_1}(\mathbf{b}_2)$ and so on.
- $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \tilde{\mathbf{b}}_j$.

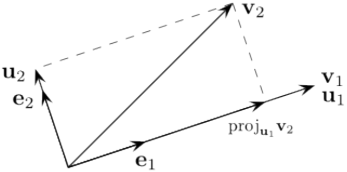
Gram-Schmidt Ortogonalization



Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$
- Output: $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$, such that $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0, i \neq j$
- Define the projection operator: $proj_u(v) = \frac{\langle v,u \rangle}{\langle u,u \rangle} u$.
- $\tilde{\mathbf{b}}_1 = \mathbf{b}_1, \tilde{\mathbf{b}}_2 = \mathbf{b}_2 - proj_{\tilde{\mathbf{b}}_1}(\mathbf{b}_2)$ and so on.
- $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \tilde{\mathbf{b}}_j$.
- Cool! Now plug-in a lattice and find an orthogonal basis!

Gram-Schmidt Ortogonalization

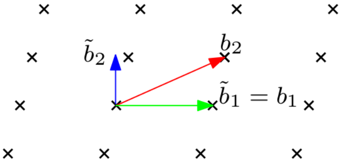


Ortogonalizations of 2 vectors in $\mathbf{R}^2$; source: Wiki

- Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$
- Output: $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \dots, \tilde{\mathbf{b}}_n$, such that $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0, i \neq j$
- Define the projection operator: $proj_u(v) = \frac{\langle v,u \rangle}{\langle u,u \rangle} u$.
- $\tilde{\mathbf{b}}_1 = \mathbf{b}_1, \tilde{\mathbf{b}}_2 = \mathbf{b}_2 - proj_{\tilde{\mathbf{b}}_1}(\mathbf{b}_2)$ and so on.
- $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \tilde{\mathbf{b}}_j$.
- Cool! Now plug-in a lattice and find an orthogonal basis! What is wrong with this approach?

Gram-Schmidt for Lattices - LLL Reduction

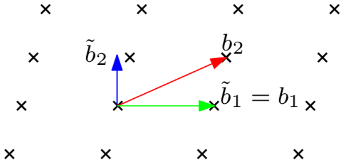
By changing the basis, we change the spanned lattice.



Ortogonalizations of 2 lattice vectors in $\mathbf{R}^2$; source: Regev O. course

Gram-Schmidt for Lattices - LLL Reduction

By changing the basis, we change the spanned lattice.



Ortogonalizations of 2 lattice vectors in $\mathbf{R}^2$; source: Regev O. course

Solution: Round the projection to the nearest integer!

Gram-Schmidt for Lattices - LLL Reduction

$\frac{3}{4}-$LLL Reduced basis

A basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is called LLL-reduced if and only if:

Gram-Schmidt for Lattices - LLL Reduction

$\frac{3}{4}$−LLL Reduced basis

A basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is called LLL-reduced if and only if:

1. $\forall i \neq j$, $|\mu_{i,j}| \leq \frac{1}{2}$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$.

Gram-Schmidt for Lattices - LLL Reduction

$\frac{3}{4}$−LLL Reduced basis

A basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is called LLL-reduced if and only if:

1. $\forall i \neq j$, $|\mu_{i,j}| \leq \frac{1}{2}$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$.
2. $\forall 1 \leq i \leq n$, $\frac{3}{4} \left\| \tilde{b}_i \right\|^2 \leq \left\| \mu_{i+1,i} \tilde{b}_i + \tilde{b}_{i+1} \right\|^2$

Why these conditions?

Gram-Schmidt for Lattices - LLL Reduction

$\frac{3}{4}$−LLL Reduced basis

A basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is called LLL-reduced if and only if:

1. $\forall i \neq j$, $|\mu_{i,j}| \leq \frac{1}{2}$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$.
2. $\forall 1 \leq i \leq n$, $\frac{3}{4} \left\| \tilde{b}_i \right\|^2 \leq \left\| \mu_{i+1,i} \tilde{b}_i + \tilde{b}_{i+1} \right\|^2$

Why these conditions?

1. Used to prove that the algorithm runs in polynomial time.

Gram-Schmidt for Lattices - LLL Reduction

$\frac{3}{4}$−LLL Reduced basis

A basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is called LLL-reduced if and only if:

1. $\forall i \neq j$, $|\mu_{i,j}| \leq \frac{1}{2}$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$.
2. $\forall 1 \leq i \leq n$, $\frac{3}{4} \left\| \tilde{b}_i \right\|^2 \leq \left\| \mu_{i+1,i} \tilde{b}_i + \tilde{b}_{i+1} \right\|^2$

Why these conditions?

1. Used to prove that the algorithm runs in polynomial time.
2. The vector b_{i+1} is not too shorter that b_i.

Gram-Schmidt for Lattices - LLL Reduction

LLL Reduction

Input: Basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$
Output: $\frac{3}{4}$-LLL reduced basis.

Algorithm 1 LLL Algorithm

```
Reduction Step:
for $i = 1$ to $N$ do
    for $j = i - 1$ to 1 do
        $b_i = b_i - \lfloor c_{i,j} \rceil b_j,\ c_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$
    end for
    Swap Step:
    if $\exists i$ s.t. $\frac{3}{4} \left\| \tilde{b}_i \right\|^2 > \left\| \mu_{i+1,i} \tilde{b}_i + \tilde{b}_{i+1} \right\|^2$ then
        Swap $b_i, b_{i+1}$; goto Reduction Step
    end if
end for
```

LLL Algorithm

1982

Many cryptosystems were broken. Every cryptosystem based on lattices must be insecure.

LLL Algorithm

1982

Many cryptosystems were broken. Every cryptosystem based on lattices must be insecure.

1996

Lattice problems are *NP*-Hard. Reductions from worst case to average case give strong security proofs.

LLL Algorithm

1982

Many cryptosystems were broken. Every cryptosystem based on lattices must be insecure.

1996

Lattice problems are *NP*-Hard. Reductions from worst case to average case give strong security proofs.

Hard problems in crypto

Cryptography requires that underlying problems are hard to solve on average, i.e. from a specific distribution

LLL Algorithm

Sad fact about LLL - in theory

It can approximate the shortest vector by an exponential factor, namely by 1.075^n.

LLL Algorithm

Sad fact about LLL - in theory

It can approximate the shortest vector by an exponential factor, namely by 1.075^n.

Interesting fact about LLL - in practice

It can approximate the shortest vector by a smaller factor 1.022^n - still exponential.

LLL Algorithm

Sad fact about LLL - in theory

It can approximate the shortest vector by an exponential factor, namely by 1.075^n.

Interesting fact about LLL - in practice

It can approximate the shortest vector by a smaller factor 1.022^n - still exponential.
In 2011 Chen and Nguyen showed that in practice you can approximate the shortest vector by 1.005^n with a variant of LLL.

Example of Hard Problems based on Lattices

Shortest Vector Problem (*SVP*)

Given an arbitrary lattice basis **B** of a n dimensional lattice $\mathcal{L}$ output a shortest non-zero lattice vector, $v \in \mathcal{L} - \{0\}$ for which $\|v\| = \lambda_1(\mathcal{L})$.

Example of Hard Problems based on Lattices

Shortest Vector Problem (*SVP*)

Given an arbitrary lattice basis **B** of a n dimensional lattice $\mathcal{L}$ output a shortest non-zero lattice vector, $v \in \mathcal{L} - \{0\}$ for which $\|v\| = \lambda_1(\mathcal{L})$.

Approximate Shortest Vector Problem (SVP_γ)

Given an arbitrary lattice basis **B** of a n dimensional lattice $\mathcal{L}$ output a shortest non-zero lattice vector bounded by a polyonimal function in n, i.e. $\|v\| \leq \gamma(n)\lambda_1(\mathcal{L})$.

Example of Hard Problems based on Lattices

Shortest Vector Problem (*SVP*)

Given an arbitrary lattice basis **B** of a n dimensional lattice $\mathcal{L}$ output a shortest non-zero lattice vector, $v \in \mathcal{L} - \{0\}$ for which $\|v\| = \lambda_1(\mathcal{L})$.

Approximate Shortest Vector Problem (SVP_γ)

Given an arbitrary lattice basis **B** of a n dimensional lattice $\mathcal{L}$ output a shortest non-zero lattice vector bounded by a polyonimal function in n, i.e. $\|v\| \leq \gamma(n)\lambda_1(\mathcal{L})$.

Approximate Decisional SVP ($GapSVP_\gamma$)

Given a lattice basis $\mathcal{B}$ of n dimensional lattice $\mathcal{L}$ for which $\lambda_1(\mathcal{L}) \leq 1$ or $\lambda_1(\mathcal{L}) > \gamma(n)$ decide which is the case.

Example of Hard Problems based on Lattices

Approximate Bounded Distance Decoding (BDD_γ)

Given a lattice basis $\mathcal{B}$ and a vector $t \in \mathbf{R}^n$ find the unique vector $v \in \mathcal{L}$ s.t. $dist(v, t) \ \leq \lambda_1(\mathcal{L})/2\gamma(n)$.

Example of Hard Problems based on Lattices

Approximate Bounded Distance Decoding (BDD_γ)

Given a lattice basis $\mathcal{B}$ and a vector $t \in \mathbf{R}^n$ find the unique vector $v \in \mathcal{L}$ s.t. $dist(v, t) \ \leq \lambda_1(\mathcal{L})/2\gamma(n)$.

Closest Vector Problem *CVP*

Currently there isn't any cryptosystem based on *CVP* - maybe because it's just too hard.

Not More Problems...

Membership Problem

Given a lattice basis $\mathbf{B} \in \mathbb{R}^{nxn}$ and a vector $v \in \mathbb{R}^n$ decide if $v \in \mathcal{L}$.

Not More Problems...

Membership Problem

Given a lattice basis $\mathbf{B} \in \mathbb{R}^{nxn}$ and a vector $v \in \mathbb{R}^n$ decide if $v \in \mathcal{L}$.

Equivalence Problem

Given 2 lattice bases $\mathbf{B}_1 \in \mathbb{R}^{nxn}$, $\mathbf{B}_2 \in \mathbb{R}^{nxn}$ decide if $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$.

Not More Problems...

Membership Problem

Given a lattice basis $\mathbf{B} \in \mathbb{R}^{nxn}$ and a vector $v \in \mathbb{R}^n$ decide if $v \in \mathcal{L}$.

Equivalence Problem

Given 2 lattice bases $\mathbf{B}_1 \in \mathbb{R}^{nxn}$, $\mathbf{B}_2 \in \mathbb{R}^{nxn}$ decide if $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$.

Something Wrong?

Not More Problems...

Membership Problem

Given a lattice basis $\mathbf{B} \in \mathbb{R}^{nxn}$ and a vector $v \in \mathbb{R}^n$ decide if $v \in \mathcal{L}$.

Equivalence Problem

Given 2 lattice bases $\mathbf{B}_1 \in \mathbb{R}^{nxn}$, $\mathbf{B}_2 \in \mathbb{R}^{nxn}$ decide if $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$.

Something Wrong?

Those are easy problems!

Ideals in Rings look alike lattices

- Polynomial Ring in $Z_p[X]/(x^n + 1)$.

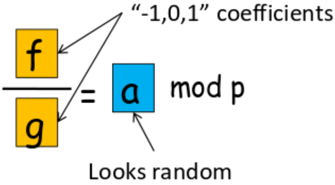
Ideals in Rings look alike lattices

- Polynomial Ring in $Z_p[X]/(x^n + 1)$.
- Elements are polynomials of degree $n - 1$ with coefficients in range $[-(p-1)/2, (p-1)/2]$. Just think about n dimensional vectors with values in Z_p.

NTRU cryptosystem

Next slides with the NTRU cryptosystem belong to Vadim Lyubashevki.

NTRU Cryptosystem

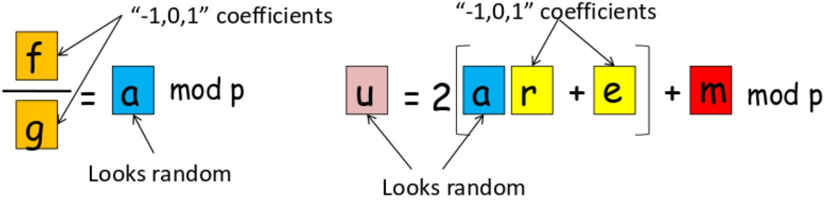


$$\frac{f}{g} = a \mod p$$

“-1,0,1” coefficients

Looks random

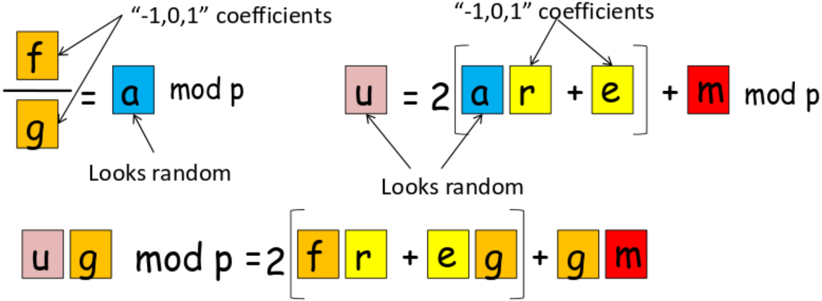
NTRU Cryptosystem



$$\frac{f}{g} = a \mod p$$

$$u = 2\left[a\,r + e\right] + m \mod p$$

NTRU Cryptosystem



“-1,0,1” coefficients

$$\frac{f}{g} = a \mod p$$

Looks random

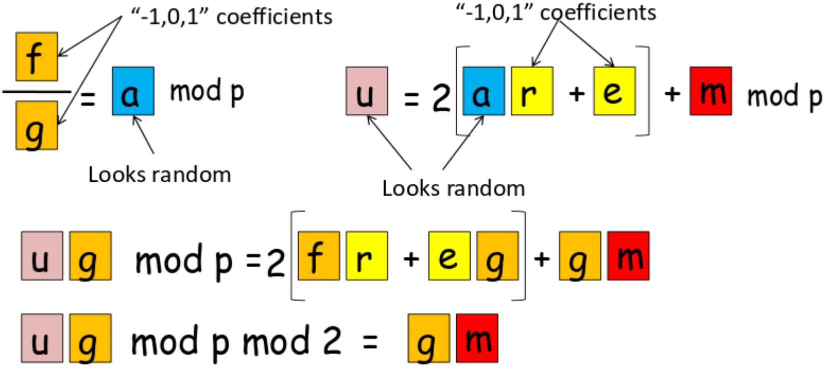
“-1,0,1” coefficients

$$u = 2\left[a\,r + e\right] + m \mod p$$

Looks random

$$u\,g \mod p = 2\left[f\,r + e\,g\right] + g\,m$$

NTRU Cryptosystem



“-1,0,1” coefficients

$$\frac{f}{g} = a \mod p$$

Looks random

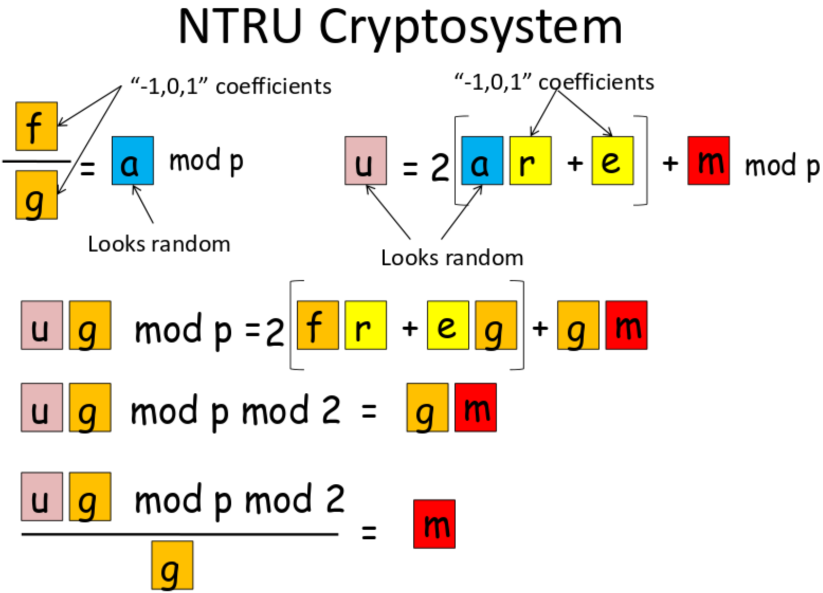
“-1,0,1” coefficients

$$u = 2\left[a\, r + e\right] + m \mod p$$

Looks random

$$u\, g \mod p = 2\left[f\, r + e\, g\right] + g\, m$$

$$u\, g \mod p \mod 2 = g\, m$$



NTRU Cryptosystem

“-1,0,1” coefficients

$$\frac{f}{g} = a \bmod p$$

Looks random

“-1,0,1” coefficients

$$u = 2\left[a\,r + e\right] + m \bmod p$$

Looks random

$$u\,g \bmod p = 2\left[f\,r + e\,g\right] + g\,m$$

$$u\,g \bmod p \bmod 2 = g\,m$$

$$\frac{u\,g \bmod p \bmod 2}{g} = m$$

Facts about NTRU

Security proofs

Until 2011 there was no proof of NTRU security. The proof is based on the hardness of Ring-LWE distribution.

Thank you!