Lecture 14:

Lattices & Short Integer Solutions

CS355 - Spring 2019
Henry Corrigan - Gibbs
May 15, 2019

Logistics

* HW 4 out now! Due Friday, May 24 @ 5pm

  Many good problems!

* Katy's OH moving to Gates B21.

* Security Seminar: "Message franking" - Paul Grubbs (Cornell)
  Friday, May 24, 4:15pm
  Gates 463A

* Event: My defense...
  "Protecting Privacy by Splitting Trust"
  Friday, May 31 1pm - 2pm
  Packard 101

Plan

* Recap

* Lattice-based crypto

* Short integer solutions

* Collision resistance from SIS

Recap: Pairing-Based Crypto

Groups $\mathbb{G}$, $\mathbb{G}_T$ of prime order q.

↳ Dlog is hard in $\mathbb{G}$ and $\mathbb{G}_T$

Pairing $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

↳ efficient, non-trivial, bilinear

$$e(g^x, g^y) = e(g,g)^{xy}.$$

Intuition:

In "normal" dlog-hard groups, can compute degree-one (linear) fns in the exponent"

$$g^x, g^y \longrightarrow g^{ax+by+c}$$

In groups w/ pairing, can compute degree-two (quadratic fns) "in the exponent"

$$g^x, g^y \longrightarrow e(g,g)^{axy+bx^2+cy^2+dx+ey+f}$$

$\Rightarrow$ Amazing applications. And practical!

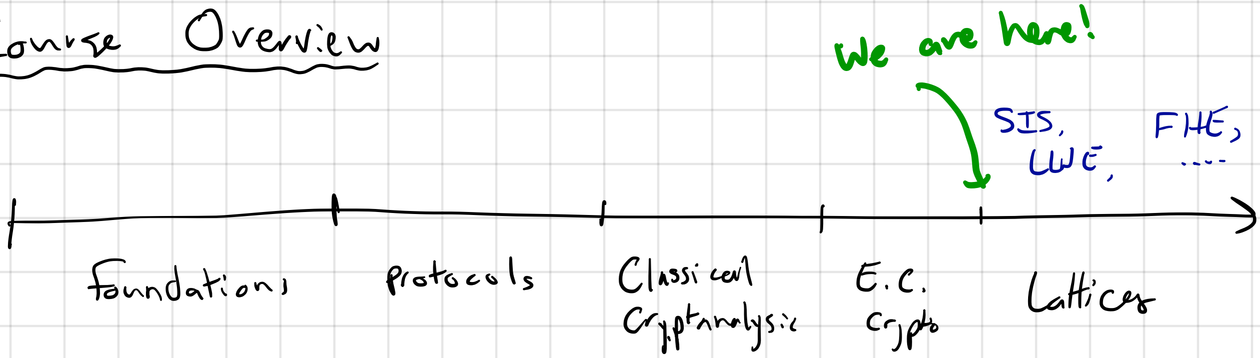
- Short sigs
- IBE
- Broadcast encryption
- Certain beautiful "SNARK" constructions $\left\{ \begin{array}{l} \text{Gennaro} \\ \text{Gentry} \\ \text{Parno} \\ \text{Raykova} \end{array} \right.$

Is there a trilinear map? Compute degree-three fns in exponent? With dlog hard in source group

$$g^x, g^y \longrightarrow e(g,g)^{f(x,y)} \quad \text{for } f(\cdot,\cdot) \text{ of deg } 3?$$

➡ Major open Q in crypto.

Course Overview



In the next few lectures, we will be talking about lattice-based crypto.

Interesting b/c:

1) Gives schemes plausibly secure against quantum attacks
   - ↳ Factoring and dlog are easy (poly time) on Q.C.
   - ↳ No known eff quantum attacks on many lattice problems

   Boaz's lectures?

2) Gives new functionality (e.g. FHE)
   - ↳ Don't know how to build from number theory (DDH, RSA, ...)
   - ↳ Next week

3) Nice theoretical consequences
   - ↳ Base crypto on worst-case hardness

N.B. There are people who work on lattice-based crypto for each of these three reasons.

NIST is standardizing new P.Q. crypto schemes now... many based on lattice constructions.

N.B. This lecture based on Peikert's lattice survey and lecture notes by David Wu and Sam Kim.

We will see

- Short integer soln problem (SIS) ← today
  ↳ OWFs, CRHFs, symmetric-key primitives
- Learning with errors problem (LWE) ← Next week
  ↳ PKE, IBE, FHE, ...

A warm-up problem: Subset Sum (Modular)

Input: A set of m integers

$$\langle a_1, a_2, \ldots, a_m \rangle \in \mathbb{Z}_q^m$$

Output: A non-empty subset that sums to zero (mod q)

$$\langle x_1, x_2, \ldots, x_m \rangle \in \{-1, 0, 1\}$$

Wlog to allow negatives

$$\text{s.t.} \quad \sum_{i=1}^{m} a_i x_i = 0 \in \mathbb{Z}_q$$

→ For ceratain settings of parameters (not those useful for crypto), subset sum is NP complete!

Example

Input: $\langle 10, 3, -2, 7, 4, 15 \rangle$

Output: $\langle 1, -1, 0, -1, 0, 0 \rangle$

$$\implies 10 - 3 - 7 = 0 \quad ✓$$

Short Integer Solution

A slight generalization of subset sum.

Idea: Take sums of vectors instead of single ints.

Input: m vectors $\left\langle \begin{pmatrix} | \\ a_1 \\ | \end{pmatrix}, \begin{pmatrix} | \\ a_2 \\ | \end{pmatrix}, \ldots, \begin{pmatrix} | \\ a_m \\ | \end{pmatrix} \right\rangle$ each in $\mathbb{Z}_q^n$

Output: a vector $\vec{x} \in \{-B, \ldots, -1, 0, 1, \ldots, B\}^m \subseteq \mathbb{Z}^m$

s.t. $\sum_{i=1}^{m} a_i x_i = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{Z}_q^n$ and $\vec{x} \neq \vec{0}$

Parameters:

$$\left.\begin{array}{l} n = \text{dimension of vectors} \\ m = \#\text{ of vectors} \\ q = \text{modulus} \\ B = \text{bound on sol'n size} \end{array}\right\} SIS(n, m, q, B)$$

Our modular subset sum problem is just SIS with $n=1$!

Short Integer Solution

To save space & time, we use matrix notation

Input: $A \in \mathbb{Z}_q^{n \times m}$

ℓ_∞-norm: $\max_i |x_i|$
Often defined using other norms (e.g. Euclidean)...

Output: $x \in \mathbb{Z}^m$ s.t. (1) $\|x\|_\infty \leq B$
(2) $Ax = \vec{0} \in \mathbb{Z}_q^n$.
(3) $x \neq \vec{0} \in \mathbb{Z}^m$.

As n grows, problem gets harder.
Intuition: more constraints to satisfy.

Typically, we set m, n, q, B all $\text{poly}(\lambda)$.

→ Relation b/w parameters is crucial for hardness.

Another way to think about SIS:
You're given a system of n linear equations in m unknowns modulo q. } $m \gg n$.

Your task is to find a solution to this set of equations that is small.

→ Gaussian elimination will not give you small sol'ns.

Q: For a random A, how do we even know that an SIS sol'n exists?

A: By pigeonhole!

There are 2^m choices of $x \in \{0,1\}^m$
Then each Ax takes on one of q^n values in $\mathbb{Z}_q^n$.
If

$$2^m > q^n$$

we must have an x, x' with $x \neq x'$ s.t.

$$Ax = Ax' \in \mathbb{Z}_q^n$$

$$A(x - x') = \vec{0} \in \mathbb{Z}_q^n$$

Notice:
(1) $\|x - x'\| \leq 1$
(2) $A(x - x') = 0$
(3) $x - x' \neq \vec{0}$

$\to$ $x - x'$ is an SIS sol'n!

$\Rightarrow$ If we take $m > n \log q$, there must be a solution. (Generalizes to larger $B > 1$.)

Applications of SIS

* On your HW, you'll show how to construct a OWF from SIS
  - By implications at start of course, this gives PRG, PRF, PRP, MAC, Signatures, ...
    - Often called "minicrypt" primitives. See Impagliazzo's "Five Worlds" Paper.

* We can also construct CRHFs from SIS.
  - So clean, so slick!
  - If you were stuck on a desert island and needed a CRHF, this is what you'd use!

Defn Collision-Resistant Hash Fn (CRHF) [Drawn from David Wu's notes]

A keyed fn family $H: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a CRHF if

(1) It's compressing: $|\mathcal{X}| > |\mathcal{Y}|$

(2) It's collision resistant. $\forall$ eff adv Adv:

$$\Pr\left[\begin{array}{c} H(k,x) = H(k,x') \\ x \neq x' \end{array} : \begin{array}{c} k \xleftarrow{R} \mathcal{K} \\ (x,x') \leftarrow \text{Adv}(k) \end{array}\right] < \text{negl}$$

For simplicity, I left the sec param λ implicit.

CRHF from SIS

Let n, m, q be params s.t. $SIS(n,m,q,1)$ is hard.

$\mathcal{K} = \mathbb{Z}_q^{n \times m}$, $\mathcal{X} = \{0,1\}^m$, $\mathcal{Y} = \mathbb{Z}_q^n$

$$\boxed{H_{SIS} : \mathbb{Z}_q^{n \times m} \times \{0,1\}^m \to \mathbb{Z}_q^n}$$

$$H_{SIS}(A, x) := A \cdot x \in \mathbb{Z}_q^n$$

So simple!

Why is this CRHF?

Say that we have adv Adv that breaks CRHF

$$\Pr\left[\begin{array}{c} H(A,x) = H(A,x') \\ x \neq x' \end{array} : \begin{array}{c} A \xleftarrow{R} \mathbb{Z}_q^{n \times m} \\ (x,x') \leftarrow Adv(A) \end{array}\right] \geq \varepsilon.$$

Then Adv solves $SIS(n,m,q,1)$!

$x - x'$ is: (1) $x, x' \in \{0,1\}^m \Rightarrow \|x - x'\|_\infty \leq 1$

(2) $H(A,x) = H(A,x') \Rightarrow Ax = Ax' \Rightarrow A(x-x') = \vec{0}$

(3) $x \neq x' \Rightarrow x - x' \neq 0$.

Break $SIS(n,m,q,1)$ w.p. ε!

Stepping back: Lattices

Why is SIS called a "lattice" problem?

Take a set of n vectors over $\mathbb{Z}^n$ ("basis")

$$B = (\vec{b}_1, \dots, \vec{b}_n).$$

Look at all linear combinations

$$\vec{v} = \sum a_i \vec{b}_i \qquad \text{for } a_1, \dots, a_n \in \mathbb{Z}$$

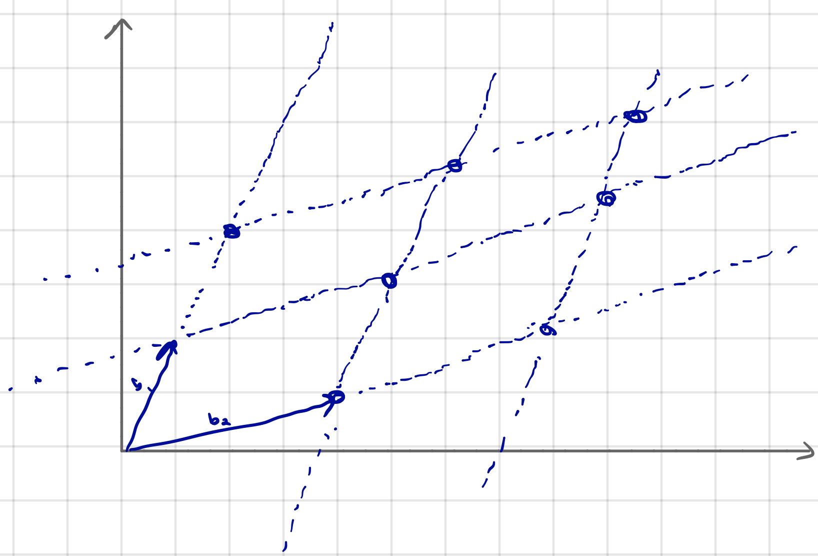
or

$$v = B \cdot \vec{a} \qquad \text{for } a \in \mathbb{Z}$$

This is the "lattice" $\mathcal{L}(B)$ generated by basis B.

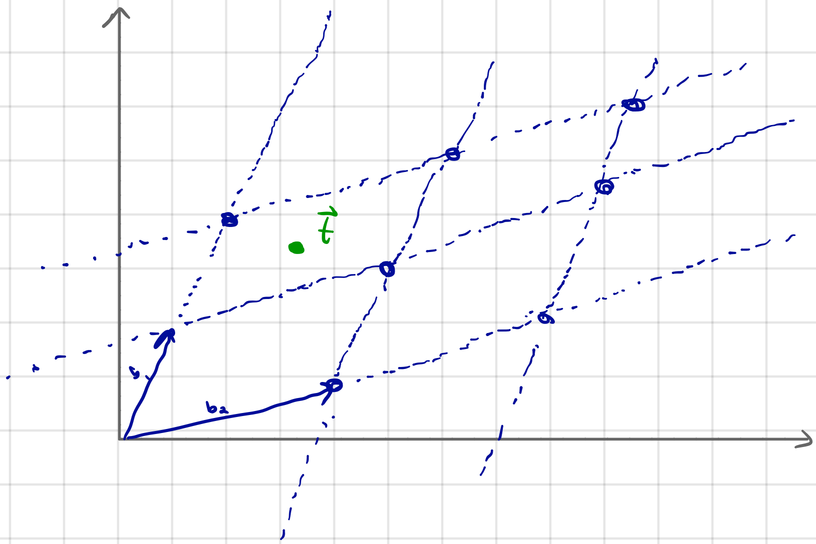
When $n=2$, this looks like a lattice

$B = \{b_1, b_2\}$



Lattices

Given a basis B for a lattice $\mathcal{L}(B)$, there are many questions you can ask?



1) Shortest vector problem (SVP)

Q: What is the shortest non-zero vector in $\mathcal{L}(B)$?

e.g. using ℓ_2-norm

→ Best algs run in super-poly time (in n).
→ In fact, this is NP hard.

2) Closest vector problem (CVP)

Q: Given basis B and "target" $\vec{t} \in \mathbb{Z}^n$, what is vector $\vec{v}$ in $\mathcal{L}(B)$ s.t. $\|\vec{t} - \vec{v}\|$ is minimized?

→ Best algs run in super-poly time (in n).

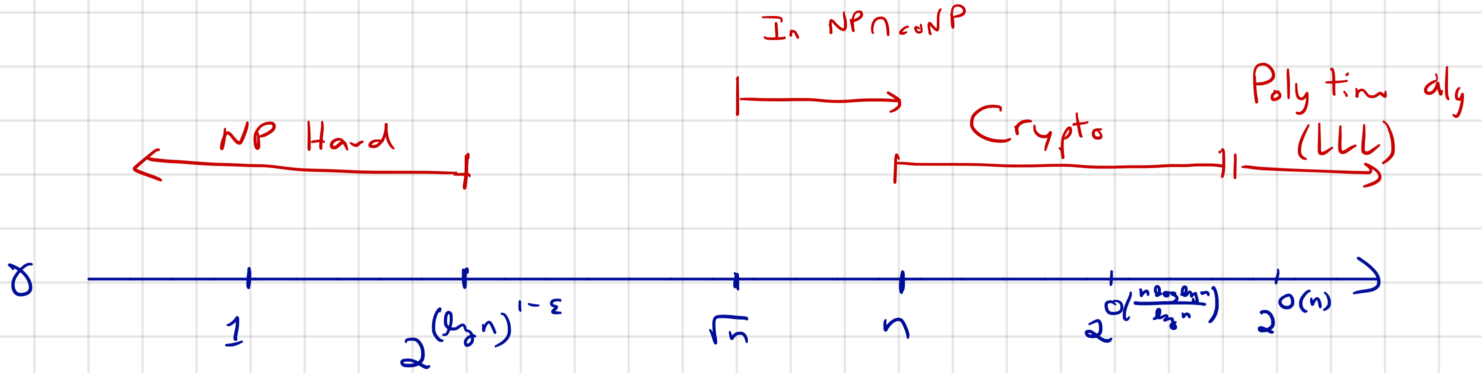
3&4) SVP_γ and CVP_γ (for $\gamma > 1$)

Solve SVP approximately... off by a factor of γ.

→ As γ grows, problem gets easier.

Hardness of SVP_γ (Vinod V's lecture notes)

Approx factor γ nicely interpolates b/w NP hard and ppt, w/ crypto in between.



Major open Q: Base crypto on NP hardness?

Relation to SIS... Ajtai (followed by many others) showed that breaking SIS $\Rightarrow$ solving certain lattice problem (GapSVP_γ) on any lattice.

"Basing crypto on worst-case hardness"

Solving $SIS(n, m, q, B)$ for $m = poly(n)$
$B > 0$
$q \geq B \cdot poly(n)$ large enough

$\Rightarrow$ Solve GapSVP_γ on arbitrary dim-n lattice w.h.p. for $\gamma = B \cdot poly(n)$.