Recent Developments in Cryptography: lattices and beyond

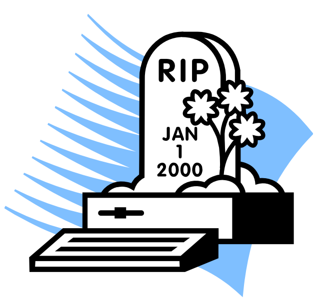
Dan Boneh

Stanford University

12th annual computer forum: Apr. 30, 2010

Lattice-based Cryptography

	Modular Arithmetic	Lattice crypto
■	enc., sigs., and beyond	enc., sigs. (recent)
■	**Performance**: slow	faster
■	**Sizes:**	
❑	public key: 20 bytes	**200 KB** (but getting better)
❑	Signatures: 20 bytes	1024 bytes
■	**Quantum computers:**	



Constructions

- Many systems from hard lattice problem:
  - Public key encryption [R'04]
  - Computing on ciphertexts (fully homomorphic encryption) [G'08]
  - Sigs. and identity based enc. [GPV'07, CHPK'10, ABB'10]
  - Searching on encrypted data

  (current constructions result in long public keys and signatures)

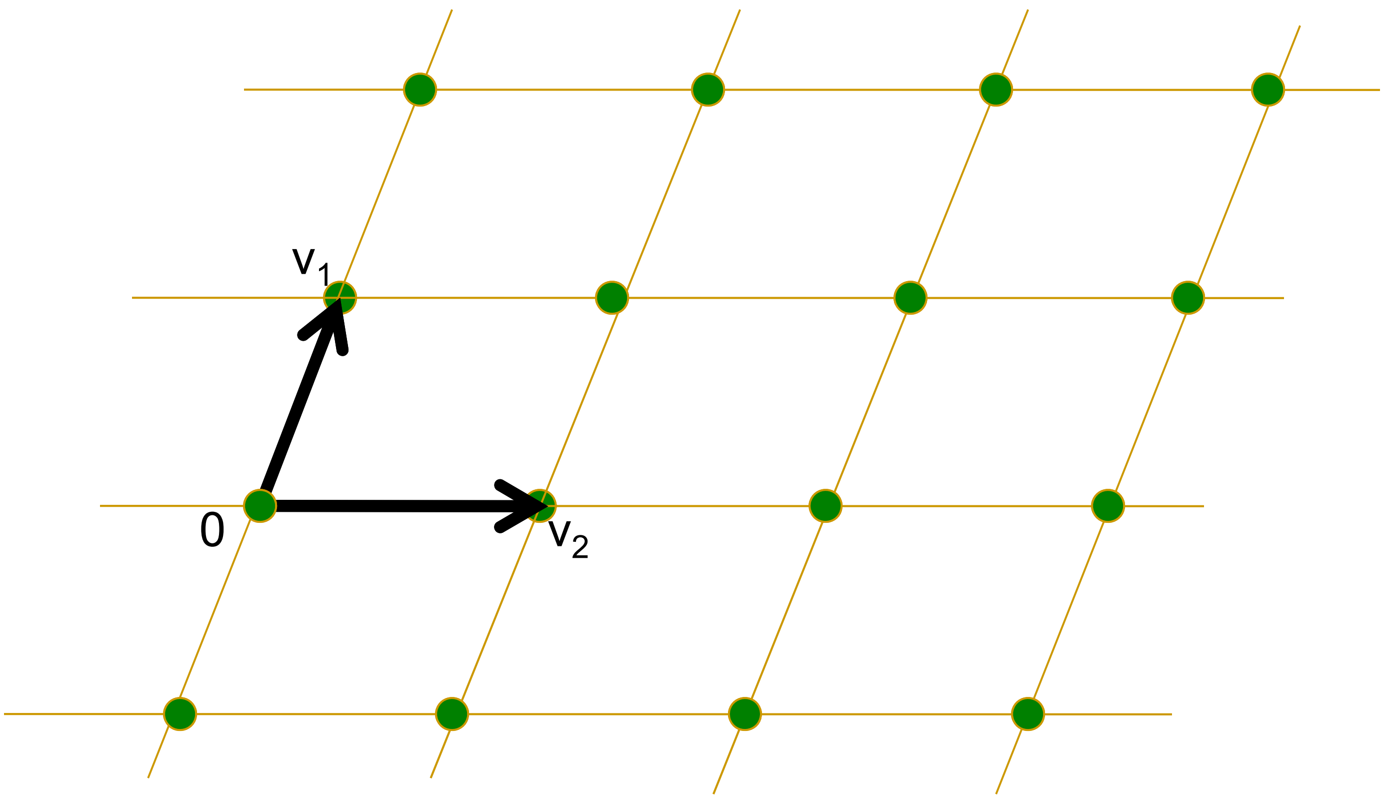
- End goal:
  secure systems w/o relying on hardness of factoring

What is a lattice in R^m ?   (e.g. $m = 512$)

- All integer linear combinations of given basis vectors

$$L(M) = \{\ M^T s \quad \text{for all} \quad s \text{ in } Z^m\ \}$$

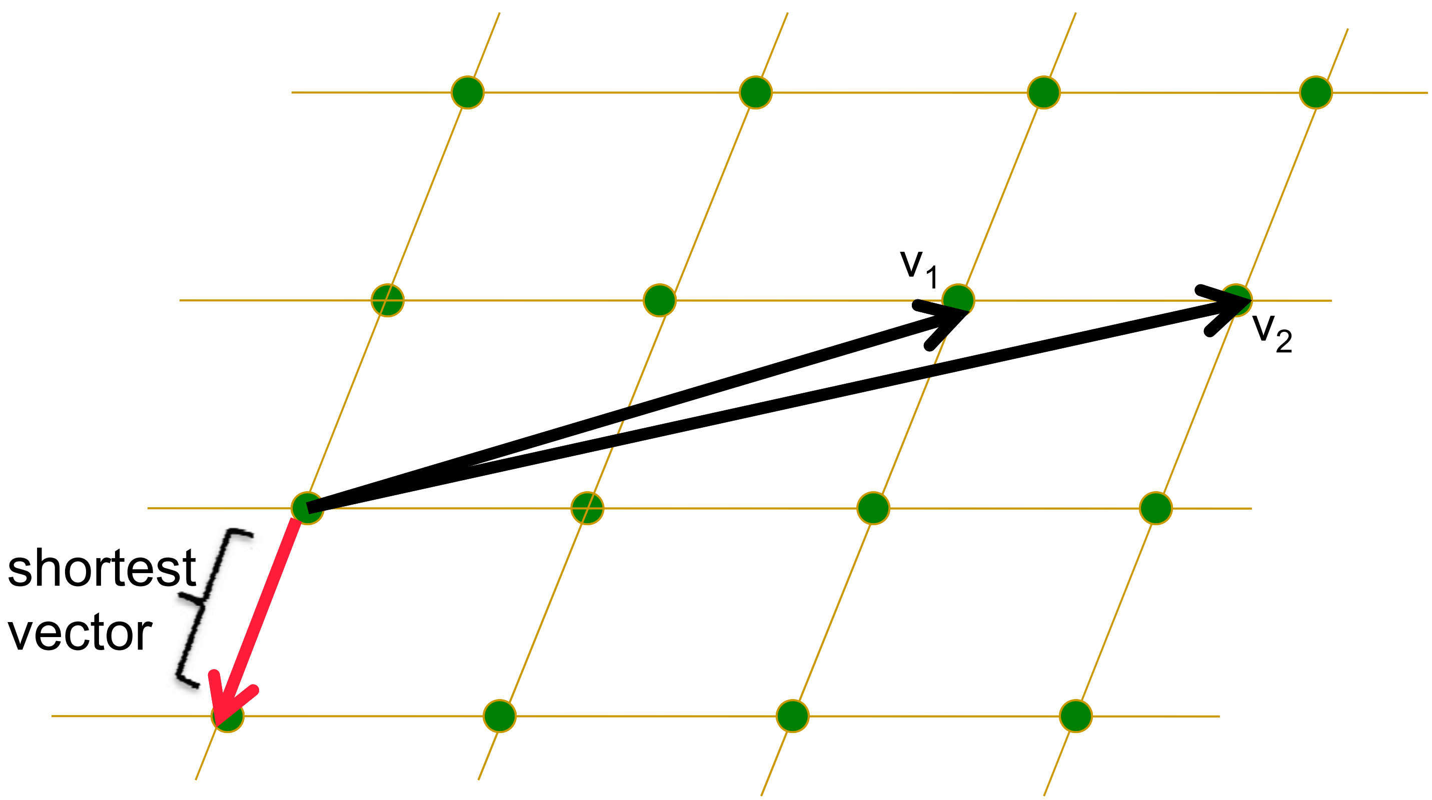
$$M = \begin{bmatrix} \text{--}\ v_1\ \text{--} \\ \dots \\ \text{--}\ v_m\ \text{--} \end{bmatrix}$$



Not all basis of L are nice

- “Bad” basis: contains “long” vectors

  (both basis span the same lattice)



Hard problems on lattices

- Hard: given a **bad** basis for L find a **short** basis for L.
  - But: can easily generate a pair (L, B) where
    L is a lattice and B is a short basis for L [A96, AP09]

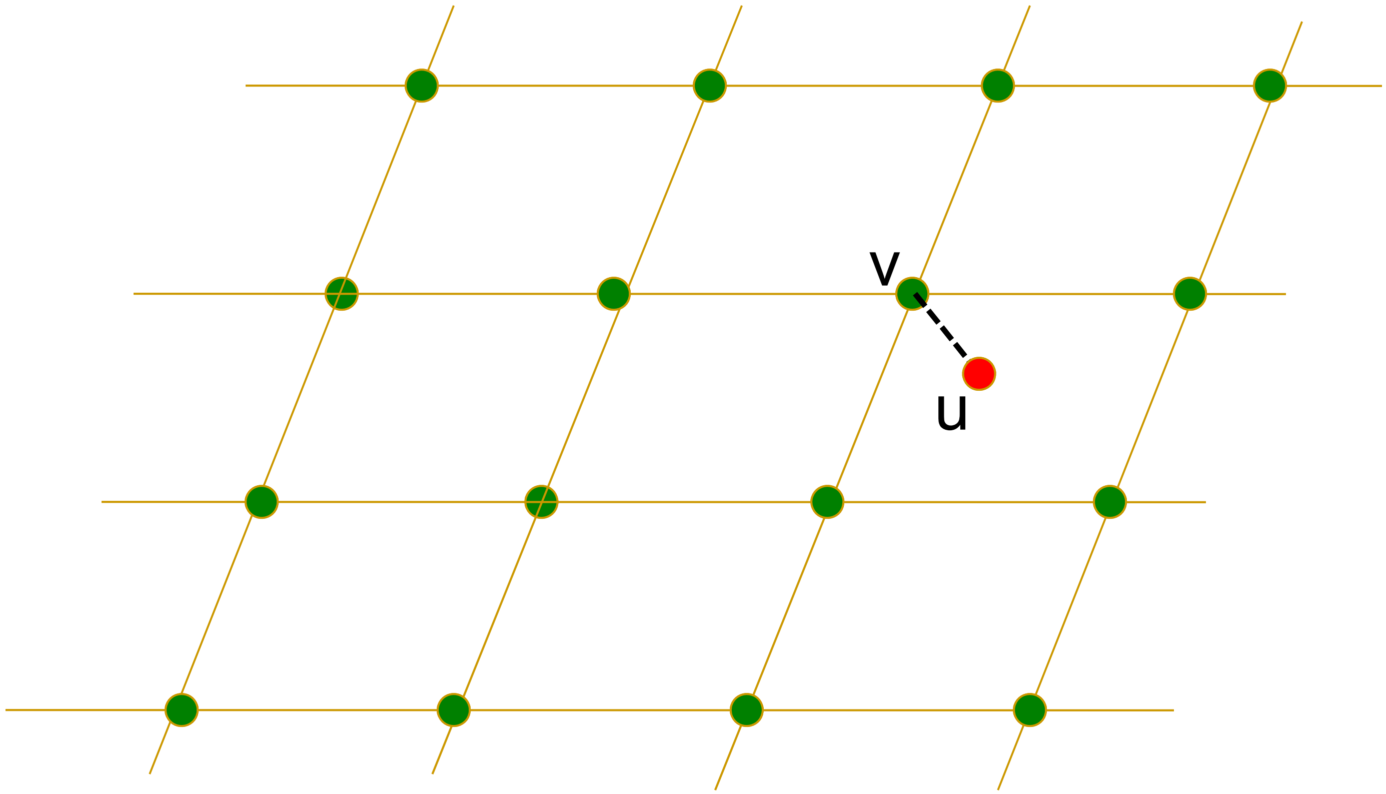
- Even finding one short vector is hard:

  SVP: given a **bad** basis for L find a "short" vector v in L

- How hard is SVP?
  - Ajtai'96: finding short vector in certain "random" lattices is as hard as finding short vector in hardest lattices

Closest vector problem

- CVP: given a **bad** basis for L(M) and u , find v



$v = M^T s$ and $u = M^T s + \Delta$

Example app: Lattice Diffie-Hellman

- Recall the basic Diffie-Hellman key exchange protocol

[group G of order q and g in G]

rand. a

rand. b

$A = g^a$

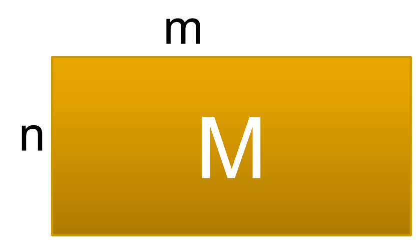
$B = g^b$

key $= B^a$

key $= A^b$

$$B^a = A^b = g^{ab}$$

Lattice Diffie-Hellman [R'04]



- Public: random matrix M in $(Z_q)^{nxm}$

rand. a in $(Z_q)^n$

rand. short Δ in $(Z_q)^m$

rand. **short** b in $(Z_q)^m$

$a' = (M^T a) + \Delta$ in $(Z_q)^m$ →

← $b' = (M\, b)$ in $(Z_q)^n$

$(b' \bullet a) = a^T M\, b$

$(a' \bullet b) = a^T M\, b + \underbrace{(\Delta^T b)}_{\text{small}}$

key = round($b' \bullet a$) = round($a' \bullet b$)

Security

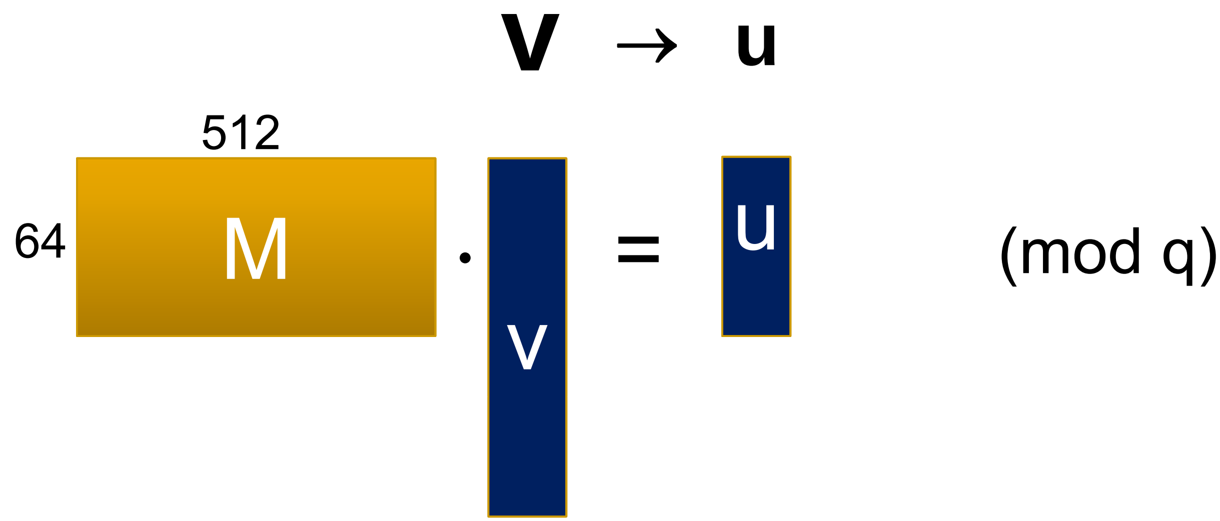
- Eavesdropper sees M , a’ , b’ and wants key
- Regev’04:
  - recovering key from M, a’, b’ is as hard as

    finding short vectors on hardest lattices

    (on a quantum computer)

- Note: can derive two lattice public-key encryption systems from lattice Diffie-Hellman

The ISIS problem



The ISIS problem: (hard)

given **M, u** find **v** s.t. **M·v = u** and $\|\mathbf{v}\|_2$ **“small”**

ISIS has a trapdoor (for fixed **M**): [GPV’08]

“short” solutions to **M·v = 0** $\Rightarrow$ can solve ISIS for any **u**

Example ISIS-based Signatures [ABB'10]

- Public key: matrices A B R

  Secret key: ISIS trapdoor for A

- **Sign(msg):** define $\mathbf{M = (A \mid A{\cdot}R + msg{\cdot}B)}$

  sig = "short" $\mathbf{v}$ s.t. $\mathbf{M \cdot v = 0}$ (mod q)

|sig| = 2KB

Thm: selective forgery attack $\Rightarrow$ efficient ISIS algorithm

(no random oracles)

Summary

- Lots of structure ➔ new systems with good properties

- Many open questions:

  - How big should the lattice be?
    - How hard are these problems for real-world params?
    - What is the performance in practice?

  - Are lattice problems hard for quantum computers ?

THE END