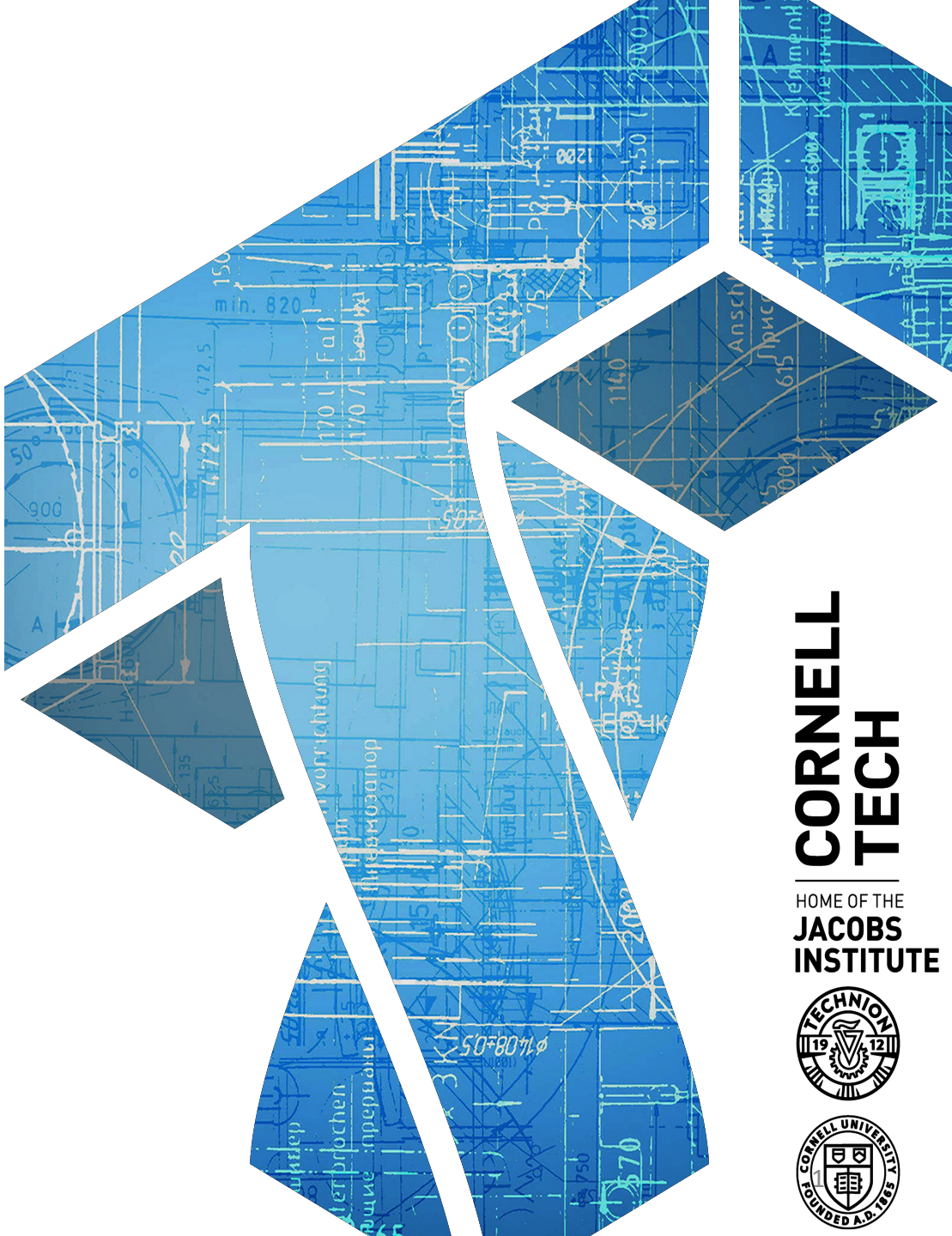
CS 5830
Cryptography



CORNELL TECH

HOME OF THE JACOBS INSTITUTE

Asymmetric crypto so far

- **RSA**
  - Work in $\mathbf{Z}_N^*$ for large composite N = pq
  - RSA assumption: given X^e mod N can't recover X without secret key d
- **Discrete log problem (DLP)**
  - Work in prime-order subgroup of $\mathbf{Z}_p^*$ for prime p. g is a generator of subgroup. q is size of subgroup
  - Discrete log assumption: given g^x mod p can't recover x
- **Elliptic curves**
  - Work in groups consisting of solutions to elliptic curve equations
  - Discrete log style
  - Can equip some with bilinear pairings
- **Today: lattice-based crypto**

Two important quantum algorithms

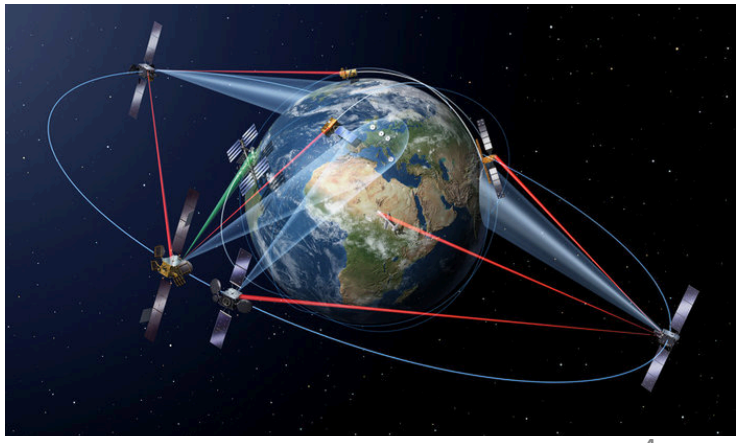
- [Shor 1994] factors composite number N
  - Recall, fastest algorithm we can implement is NFS. Runs in time
$$\mathcal{O}\left(e^{1.9(\log N)^{1/3}(\log\log N)^{2/3}}\right)$$
  - Shor's algorithm gives solution using quantum circuit of size
$$\mathcal{O}\left((\log N)^2(\log\log N)(\log\log\log N)\right)$$
  - Can be used to compute discrete logs as well

- [Grover 1996] inverts functions with quadratic speedup
  - Uses $\mathcal{O}\left(\sqrt{2^k}\right)$ to recover K from $E_K(0^n)$, for |K| = k bits
  - Double key lengths: AES-256 all good

Cryptopoclypse?

- "Post-quantum crypto" (PQC)
  - Asymmetric algorithms with conjectured security against QCs
  - Hash-based signatures, ***lattice-based***, code-based, non-linear systems of equations, elliptic curve isogenies
  - Gaining practical momentum:
    - NIST competition, practical variants of TLS key exchange

- Quantum key distribution
  - Interesting concept, but turned into snake oil
  - Run away!



Cryptopoclypse?

Why are people excited about this now?

1. Quantum computers are getting realer (lots of $$$ thrown at it)
2. Traffic encrypted *now* under possibly future-vulnerable algorithms

Evan Jeffrey's (Google's QC team) at RWC 2017:

Factoring 2048-bit RSA: 250,000,000 physical q-bits with 99.9% accuracy

State-of-the-art QC: 9 physical q-bits with 99.5% accuracy

Classical attacks may invalidate existing algorithms (including PQC!!!)

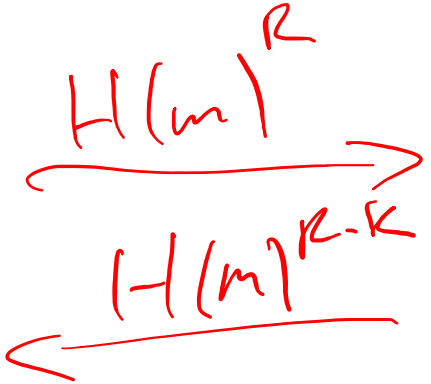
A conjecture

"Standardized PQC cryptosystems will be broken classically before quantum computers break a non-PQC cryptosystem"

Recent supporting evidence:

- Supersingular Isogeny Diffie–Hellman key exchange [Jao, De fao 2011]
- Supersingular Isogeny Key Encapsulation (SIKE, fourth round candidate for NIST PQC competition) in 2017
- [Castryk, Decru 2022], [Maino, Martindale 2022]:
  - key recovery in one hour on a single core

New asymmetric crypto is cool even forgetting about QCs



- Having diverse set of assumptions good
- Raises lots of questions about cryptographic agility
  - How do we guarantee we can move to other schemes when others are broken? Etc.
- Provide algebraic structure that's been useful for new theory
  - Fully-homomorphic encryption

Lattices

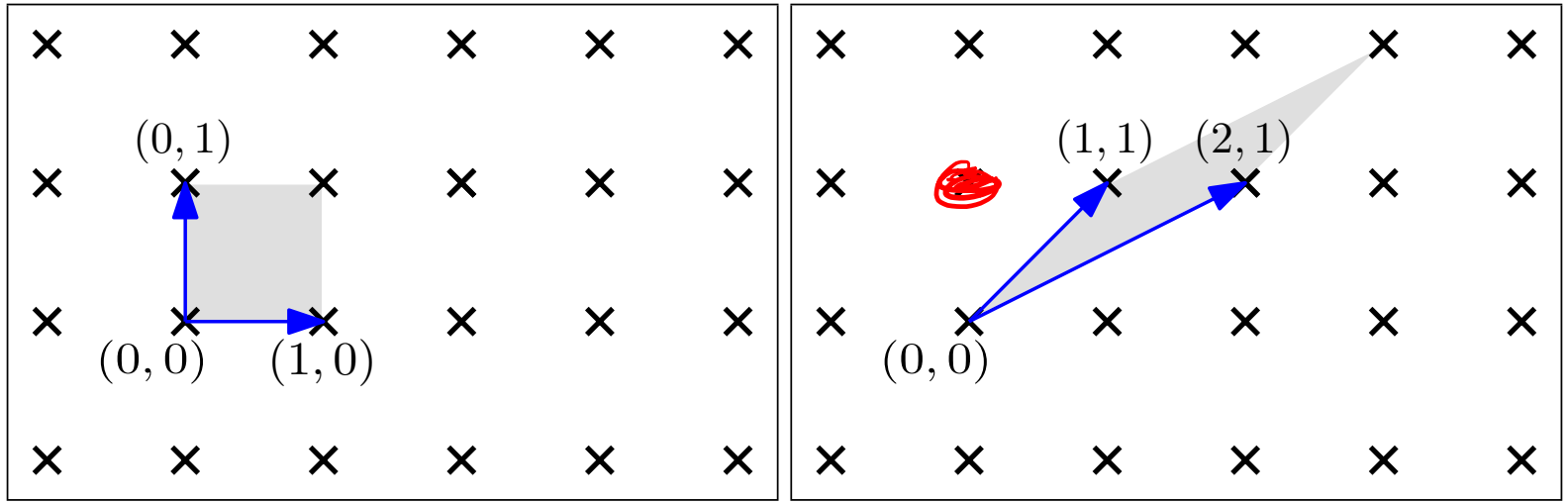
Lattices defined using basis $\mathbf{b}_1, \ldots, \mathbf{b}_m$ each in $\mathbb{R}^n$

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \left\{ \mathbf{v} = \sum_{i=1}^{m} x_i \cdot \mathbf{b}_i \ : \ x_1, \ldots, x_m \in \mathbb{Z} \right\}$$

Then $\mathcal{L} \subseteq \mathbb{R}^n$ is additive subgroup of $\mathbb{R}^n$

Crypto uses full rank lattices: m = n

Examples in n=2



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

Basis not unique!

Lattices

Lattices defined using basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ each in $\mathbb{R}^n$

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \mathbf{v} = \sum_{i=1}^{n} x_i \cdot \mathbf{b}_i \ : \ x_1, \ldots, x_n \in \mathbb{Z} \right\}$$

Shortest vector in a lattice: $\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$

Shortest Vector Problem (SVP-γ):

Given: $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^n$ Compute $\mathbf{v}$ such that $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$

SVP-γ

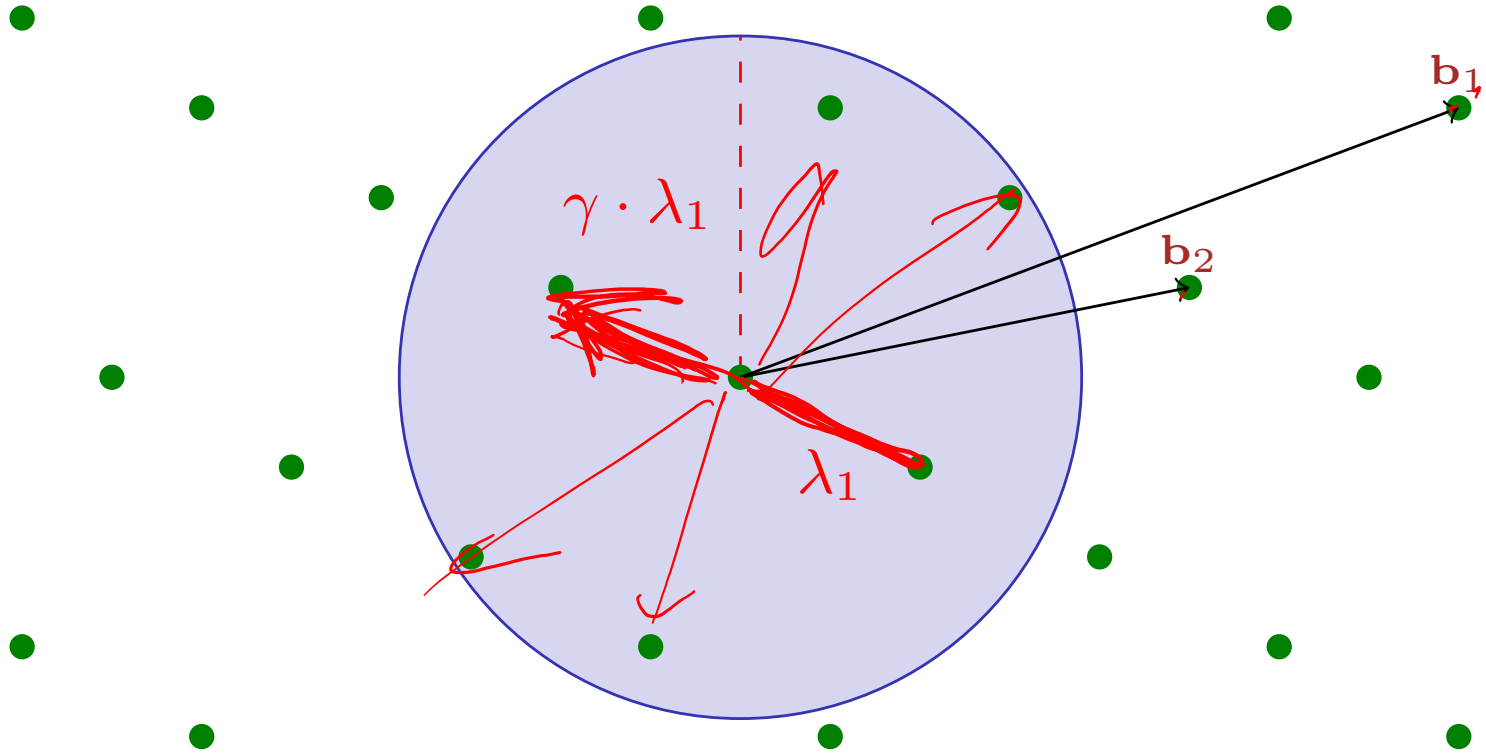


Diagram from: http://web.eecs.umich.edu/~cpeikert/pubs/slides-abit1.pdf

Lattices

Lattices defined using basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ each in $\mathbb{R}^n$

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \mathbf{v} = \sum_{i=1}^{n} x_i \cdot \mathbf{b}_i \ : \ x_1, \ldots, x_n \in \mathbb{Z} \right\}$$

Shortest vector in a lattice: $\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$

Shortest Vector Problem (SVP-γ):

Given: $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^n$ Compute $\mathbf{v}$ such that $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$

Gap Shortest Vector Problem (GapSVP-γ):

Given: $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^n$

$d \in \mathbb{R}$

Determine if: $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$

or

$\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma \cdot d$

GapSVP-γ

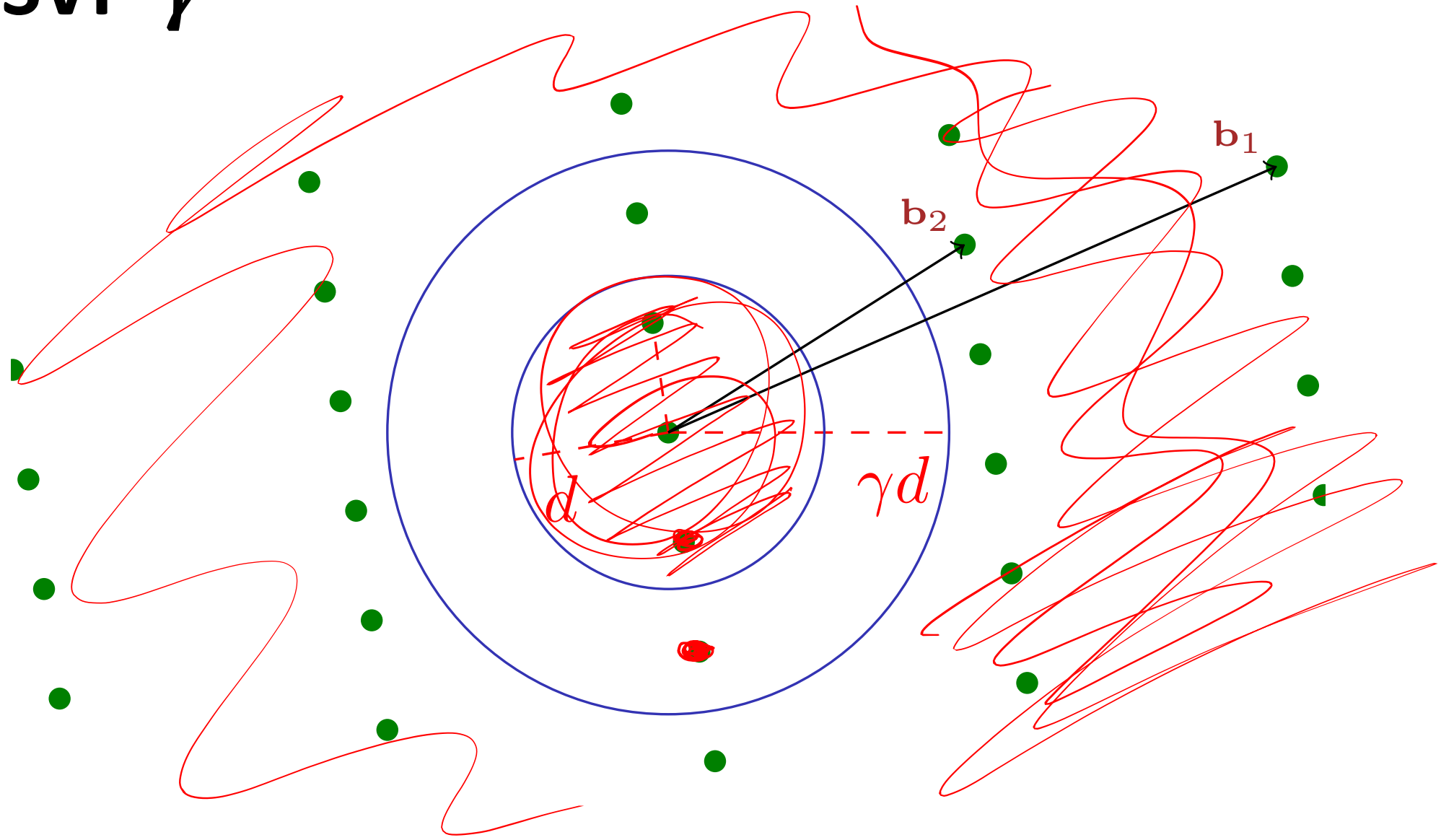


Diagram from: http://web.eecs.umich.edu/~cpeikert/pubs/slides-abit1.pdf

What is known about these problems?

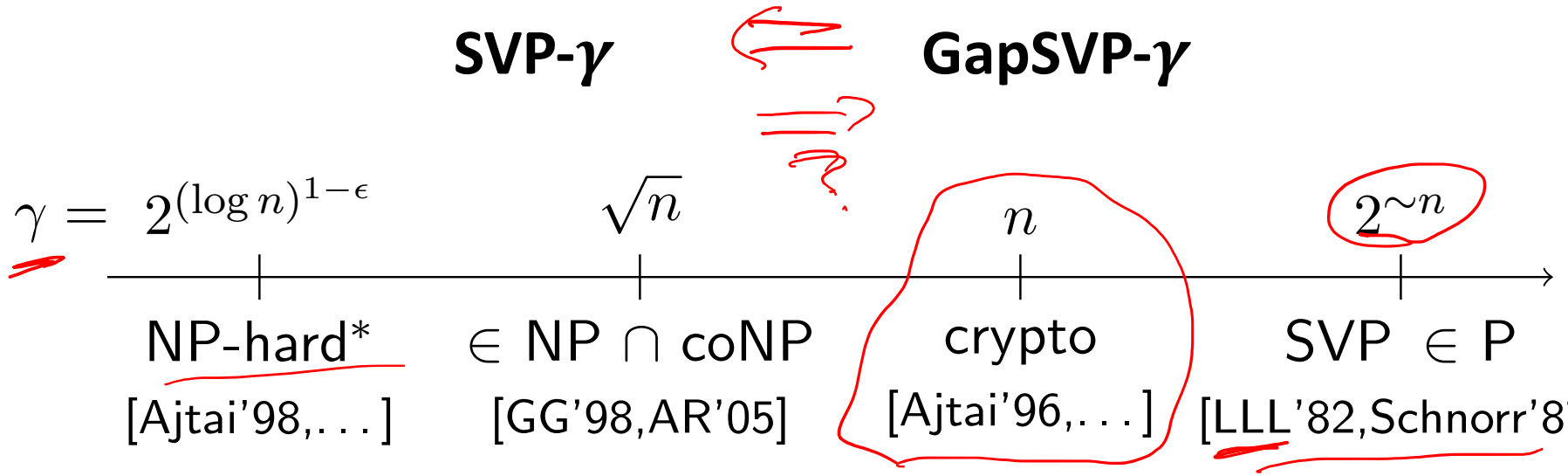


Diagram from: http://web.eecs.umich.edu/~cpeikert/pubs/slides-abit1.pdf

Best algorithms known for **SVP-γ** are exponential:

γ = poly(n) requires time 2^n

$\gamma = 2^k$ requires time $2^{n/k}$

True also for known **quantum algorithms**

Don't build cryptosystems directly from SVP problems

Instead:

Assumptions reducible to (Gap)SVP

Learning with errors (LWE)

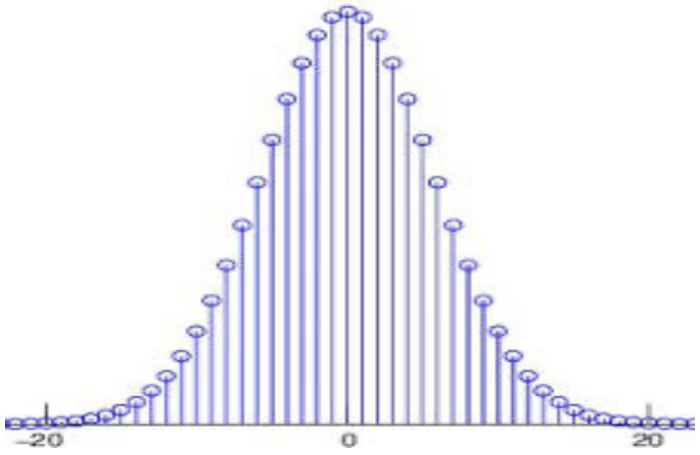
Let $\mathbf{s} \in \mathbb{Z}_q^n$ be secret vector

Let $\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{m \times n}$ be random m x n matrix

Let $\mathbf{e} \leftarrow\!\!\$\ \chi^m$ be "noise" vector

All values integers mod q!

Errors taken from "bounded" distribution (w.h.p. very small!)
Discrete Gaussian:



Given: $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ Compute: $\mathbf{s}$

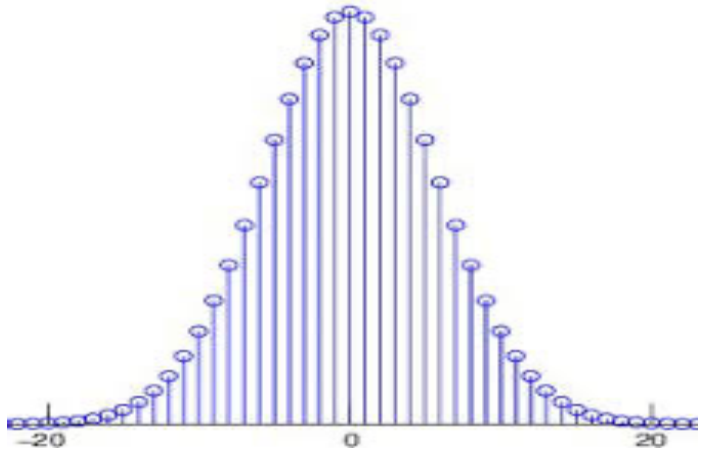
System of m linear equations with n+m unknowns

$$\begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \dots & \mathbf{a}_{1n} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \dots & \mathbf{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{m1} & \mathbf{a}_{m2} & \dots & \mathbf{a}_{mn} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_m \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix}$$

Learning with errors (LWE)

System of m linear equations with n+m unknowns

$$\begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \dots & \mathbf{a}_{1n} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \dots & \mathbf{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{m1} & \mathbf{a}_{m2} & \dots & \mathbf{a}_{mn} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix} - \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_m \end{bmatrix}$$



1) If errors are uniform over $\mathbf{Z}_q$ then finding **s** impossible
2) If error removed, sequence of m equations in n unknowns.
   - When m >= n, solvable via Gaussian elimination
3) Adding a bit of noise messes up Gaussian elimination
   - Error “adds up” when doing linear combinations of the equations

Learning with errors (LWE)

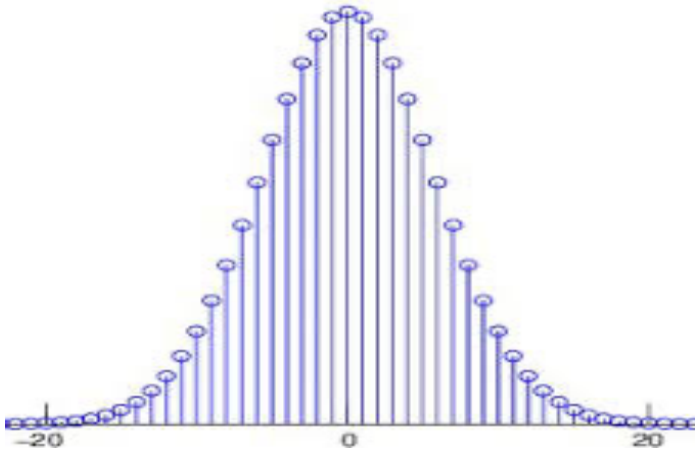
Let $\mathbf{s} \in \mathbb{Z}_q^n$ be secret vector

Let $\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{m \times n}$ be random m x n matrix

Let $\mathbf{e} \leftarrow\!\!\$\ \chi^m$ be "noise" vector

All values integers mod q!

Errors taken from "bounded" distribution (w.h.p. very small!)
Discrete Gaussian:



(Search problem) Given: $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ Compute: $\mathbf{s}$

(Decision problem)

Given: $(\mathbf{A}, \mathbf{Z}_b)$

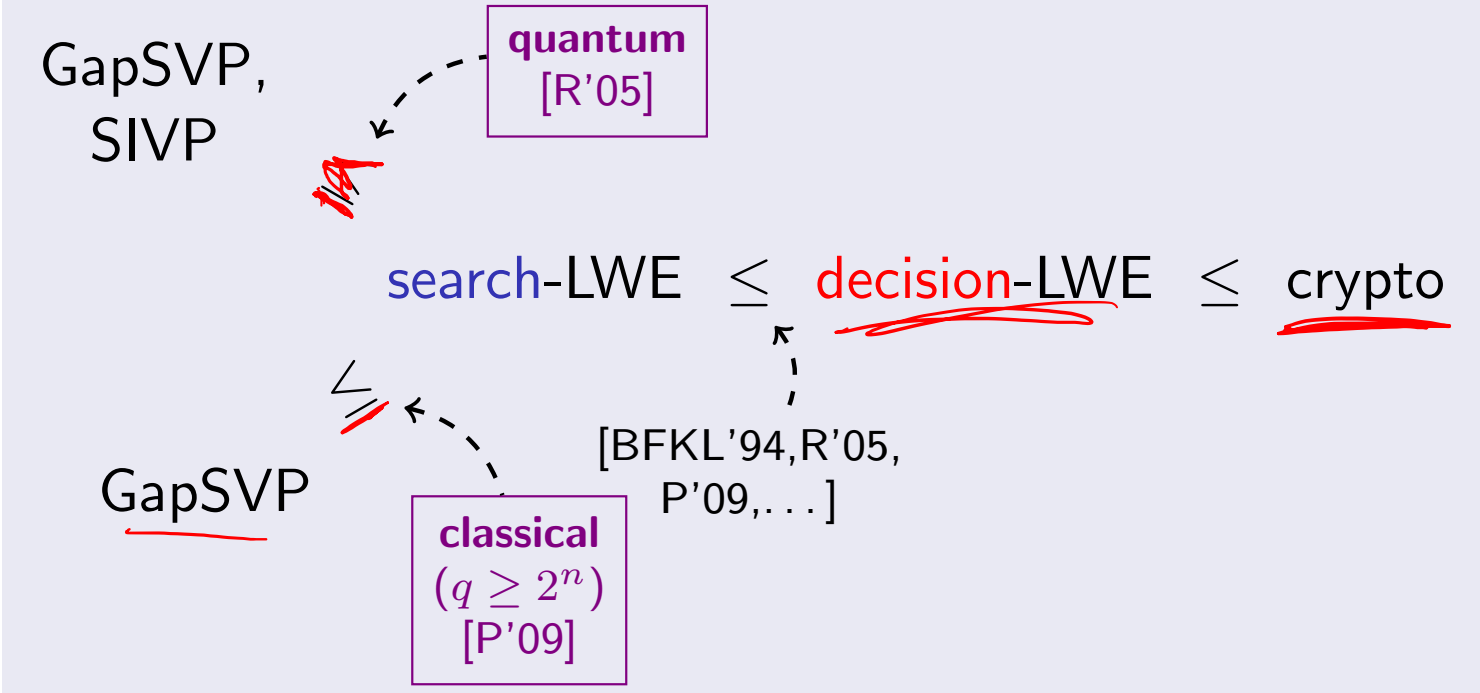
$\mathbf{Z}_1 = \mathbf{As} + \mathbf{e}$ $\mathbf{Z}_0 \leftarrow\!\!\$\ \mathbb{Z}_q^m$

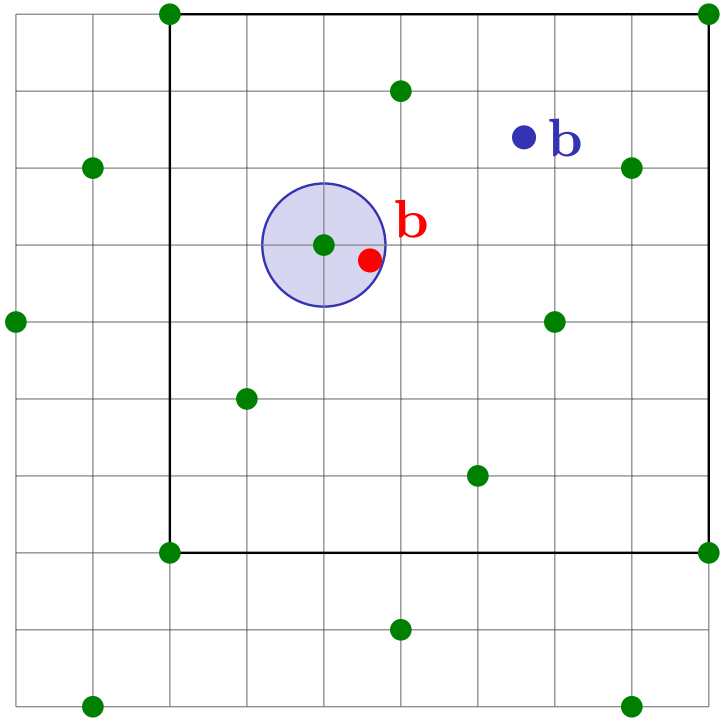
Determine b

Surprising fact: decision reducible to search problem, for same m,n,q,χ !

What is connection between LWE and lattices?

Worst-case hardness of GapSVP implies LWE (for certain ranges of parameters)





Diagrams from http://web.eecs.umich.edu/~cpeikert/pubs/slides-abit2.pdf

Regev's encryption scheme from LWE

Builds PKE for single bit messages that tightly reduces to (decisional) LWE

sk = $\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^n$

pk = $\mathbf{As} + \mathbf{e}$

A public, random m x n matrix, can be reused for keys

Encrypt bit m via: $\mathbf{u} \leftarrow\!\!\$\ \{0,1\}^m$

$$(c_1, c_2) = \left(\mathbf{u}^T\mathbf{A}\ ,\ \mathbf{u}^T\mathrm{pk} + m\lceil q/2 \rceil \right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

message

Decrypt via: $m' \leftarrow c_2 - c_1\mathbf{s}$

If *m'* closer to q/2 then Return m = 1

If *m'* closer to 0 then Return m = 0

When does it work?

$$\begin{aligned} m' &= c_2 - c_1\mathbf{s} \\ &= \mathbf{u}^T\mathrm{pk} + m\lceil q/2 \rceil - \mathbf{u}^T\mathbf{As} \\ &= \mathbf{u}^T\mathbf{As} + \mathbf{u}^T\mathbf{e} + m\lceil q/2 \rceil - \mathbf{u}^T\mathbf{As} \\ &= \mathbf{u}^T\mathbf{e} + m\lceil q/2 \rceil \end{aligned}$$

Should be smaller than q/4

Choose Gaussian appropriately

Regev's encryption scheme from LWE

Builds PKE for single bit messages that tightly reduces to (decisional) LWE

sk = $\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^n$

pk = $\mathbf{As} + \mathbf{e}$

Encrypt bit m via: $\mathbf{u} \leftarrow\!\!\$\ \{0,1\}^m$

$$(c_1, c_2) = \left(\mathbf{u}^T\mathbf{A} \ , \ \mathbf{u}^T\mathrm{pk} + m\lceil q/2 \rceil \right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

Security? Adversary knows: pk, **A**, c_1, c_2 must determine *m*.

Recall LWE assumptions:

(Search problem) Given: $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ Compute: $\mathbf{s}$

(Decision problem) Given: $(\mathbf{A}, \mathbf{Z}_b)$

$\mathbf{Z}_1 = \mathbf{As} + \mathbf{e}$ $\mathbf{Z}_0 \leftarrow\!\!\$\ \mathbb{Z}_q^m$

Determine b

Regev's encryption scheme from LWE

Builds PKE for single bit messages that tightly reduces to (decisional) LWE

sk = $\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^n$

pk = $\mathbf{A}\mathbf{s} + \mathbf{e}$

Encrypt bit m via: $\mathbf{u} \leftarrow\!\!\$\ \{0,1\}^m$

$$(c_1, c_2) = \left(\mathbf{u}^T\mathbf{A} \ , \ \mathbf{u}^T\mathrm{pk} + \hat{m}\lceil q/2 \rceil \right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

Security? Adversary knows: pk, **A**, c_1, c_2 must determine $\hat{m}$.

Sketch of reduction to decision LWE

1. Replace pk with uniform random vector **Z**, via easy reduction to decision LWE
2. Use "leftover hash lemma" to show that

$$\left(\mathbf{Z} \ , \ \mathbf{A} \ , \ (\mathbf{u}^T\mathbf{A}, \mathbf{u}^T\mathbf{Z}) \right) \approx \left(\mathbf{Z} \ , \ \mathbf{A} \ , \ (\mathbf{r}_1^T, r_2) \right)$$

Where ≈ means statistically close and

$$(\mathbf{r}_1^T, r_2) \leftarrow\!\!\$\ \mathbb{Z}_q^n \times \mathbb{Z}_q$$

3. Since r_2 independent of all else, perfectly hides message m

Towards practical schemes

- Regev's scheme not yet practical
  - Public-key's size $O(n^2)$ and ciphertexts $O(n)$ elements of $\mathbf{Z}_q$
  - Encrypts one bit at a time
- Ring-LWE by [Lyubashevsky, Peikert, Regev 2010]
  - LWE but over ring of polynomials over finite field
  - Big efficiency improvements: encrypt n-bit message all at once

NIST competition

- Goal to standardize PQC systems (signatures and encryption)
- Example:
  - NewHope-KEM concretizes LPR Ring-LWE approach

Parameter Set	$\|pk\|$	$\|sk\|$	$\|ciphertext\|$
NewHope512-CPA-KEM	928	869	1088
NewHope1024-CPA-KEM	1824	1792	2176
NewHope512-CCA-KEM	928	1888	1120
NewHope1024-CCA-KEM	1824	3680	2208

Sizes in bytes, https://newhopecrypto.org/data/NewHope_2018_06_14.pdf

Efficiency comparison

Security level	RSA (log N)	DLP in ECC groups
80	1024	160
112	2048	224
128	3072	256
256	15360	512

Security level (QC bit strength)	NewHope |pk|	NewHope |ctxt|
101	7424	8704
233	14592	17408

Cryptography in this course

Symmetric primitives

One-time pads

Stream ciphers / Pseudorandom generators
(RC4, ChaCha20, Salsa20)

Block ciphers
(AES, DES, Feistel networks)

Block cipher encryption modes
(CTR, CBC)

Cryptographic hash functions, password hashing, key derivation
(SHA256, SHA3, PBKDF, scrypt, HKDF)

Message authentication
(CBC-MAC, HMAC, Carter-Wegman MACs)

Authenticated encryption with associated data (AEAD)
(Encrypt-then-MAC, AES-GCM)

Asymmetric primitives

Public-key encryption
(Raw RSA, OAEP, ElGamal, Hybrid encryption)

Key exchange
(RSA key transport, Diffie-Hellman over ECC, X3DH)

Digital signatures
(RSA PKCS#1v2, PSS, Schnorr, DSA)

Threat models have expanded over time

- Originally:
  - Key recovery
  - (Full) plaintext recovery
  - Passive attacks
- Now:
  - Rule out leaking any information about plaintext (semantic security)
  - Chosen-ciphertext attacks assumed

And so have our basic primitives

Symmetric encryption:

Encryption-only modes -> AEAD -> Misuse-resistant AEAD -> Misuse-resistant, committing AEAD

Public-key encryption:

Trapdoor permutations (RSA) -> Randomized PKE -> CCA-secure PKE

Want primitives to provide security in as many contexts as possible.
Improves usability, robustness

Primitive	Use cases / examples	Security goals	Good schemes	Bad schemes
Block ciphers	Building block for symmetric encryption	Indistinguishable from random permutation	AES, 3DES	DES, Skipjack
Pseudorandom Functions (PRFs) / Message authentication codes (MACs)	Authenticating data with shared secret key, key derivation	Indistinguishable from random function	HMAC w/ good hash function OMAC, CMAC, PMAC	CBC-MAC without prefix-free encoding
Symmetric encryption (AEAD)	Main mechanism for encrypting data; TLS record layer, encrypting data at rest	Message confidentiality and associated data + ciphertext authenticity	Encrypt-then-MAC GCM OCB	Encryption only modes: CTR mode, CBC mode, ECB mode, RC4
Hash functions	Key derivation, PW hashing, digital signatures, HMAC	Behave like a public random function (implies coll resist, one-wayness, etc.)	SHA-256 SHA-3	MD4, MD5, SHA-1
Password-based key derivation	Password hashing, PW-based encryption	No shortcut attacks	PBKDF2, scrypt, bcrypt, argon2	Plain hash function

Primitive	Use cases / examples	Security goals	Good schemes	Bad schemes
RSA PKE	Encrypt symmetric key	No partial info on messages leaked to active attacker	RSA-OAEP w/ 2048 bit moduli	RSA-PKCS#1 v1.5, “raw” RSA < 2048 bit N
ECC PKE	Encrypt symmetric key	No partial info on messages leaked to active attacker		ElGamal by itself
Hybrid encryption	Encrypt data efficiently using recipient public key	No partial information on messages leaked; attacker can’t maul ciphertext	ECIES RSA-OAEP w/ one-time Encrypt-then-MAC scheme	Raw RSA kem, bad sym encryption (e.g., CBC mode)
Digital signatures	Authenticated key exchange, code signing	Unforgeability under chosen message attacks	ECDSA, RSA PSS	RSA-PKCS#1 v1.5
Diffie-Hellman key exchange	Establishing secure channel	Attacker can’t recover derived session key	ECC DH, Finite field DH	<< 256 bit ECC groups, << 2048 bit FF groups

How do cryptographers *evaluate* crypto?

- Cryptanalysis
  - Try to break it. Bonus points if it's an implementable attack
- Formal analyses by hand
  - Give rigorous definition of security, prove manually that scheme meets it (usually via reduction to underlying hard problem)
- Automated protocol analysis tools
  - Build software tools to analyze protocols for bugs
- Implementation analysis tools
  - Build software tools to analyze implementations

How do cryptographers *design* crypto?

- Avoid security through obscurity:
  - Public designs and evaluation
- Public competitions
  - Set out requirements document, solicit submissions and then have several years of people trying to break stuff
- Standardization processes (IETF / ISO)
  - Write down protocols as RFCs (Request for Comments)
  - Sometimes this is design-by-committee
- No single person can design good crypto. Community effort

What should to do if you need some crypto?

- Use existing, well-reviewed libraries
- If not a standard use case: Ask for help!
  - May get pointed to existing solution(s)
  - Even experts don't deploy crypto without others reviewing
  - Experts often happy to help answer questions
- Beware of Stack overflow; vet answers with experts

So many topics we did not get to

- Lightweight cryptography
- Encrypted databases
- Authenticated data structures
  - Merkle trees, append-only logs
- Blockchain
  - Authenticated data structures + consensus
- More advanced security analyses
- …

Moral character of cryptography

- *"Cryptography rearranges power."*
- Phil Rogaway's paper on moral aspects of cryptography
  - https://web.cs.ucdavis.edu/~rogaway/papers/moral-fn.pdf
- Some high level points:
  - Cryptography is inherently political
  - Modern history of field: "Chaumian" research versus Goldwasser-Micali research, cypherpunks, (lack of) academic political activisim
  - Call to action for cryptographers to embrace social responsibility
- Fascinating read, and would highly recommend!

Summary of the summary

- Cryptography is fascinating subject…
- and important:
  - Plays pivotal role in many security contexts
  - At center of power relationships

- Thanks for participating!