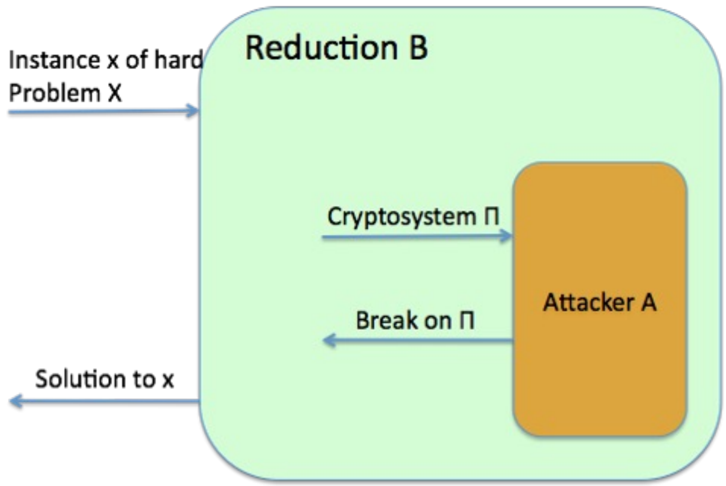
Post Quantum Cryptography: Foundations, Opportunities & Beyond

Shweta Agrawal
IIT Madras

Cryptography
The Art of Secret Keeping

Cryptography guarantees that breaking a cryptosystem is at least as hard as solving some difficult mathematical problem.



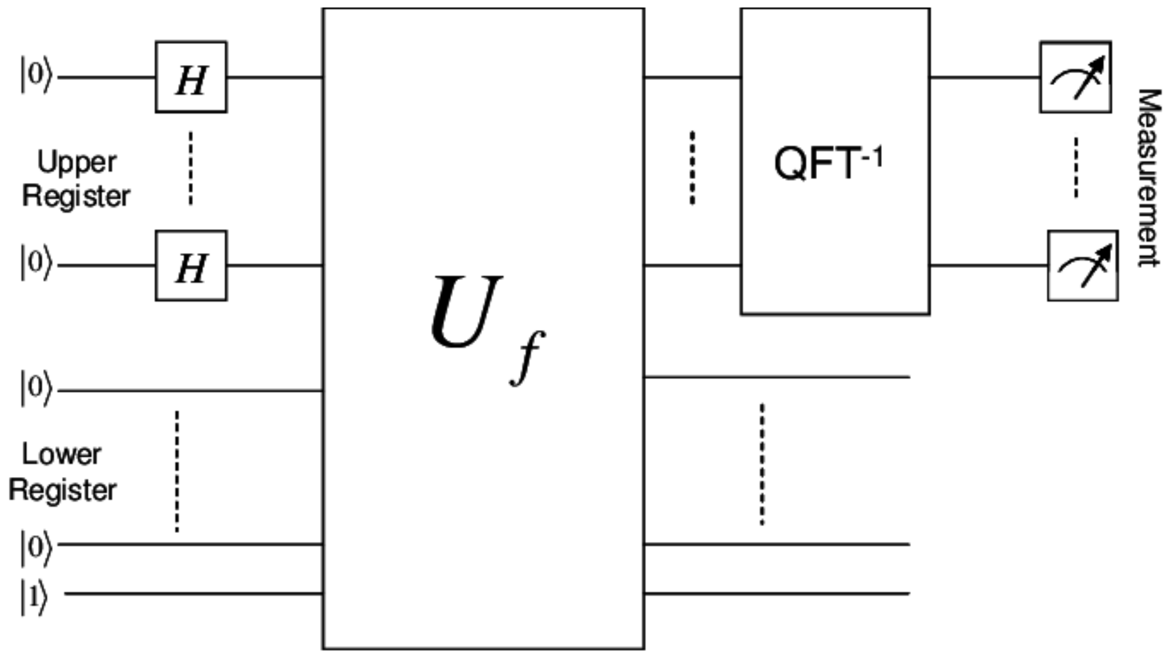
Difficult for who?

The Cryptographic Adversary

- Adversary in cryptography normally modeled by a classical computer.
- Typical guarantee is that unless the adversary can solve hard problem, attack takes more than age of universe (in CPU cycles)
- Robust to type of computer (mobile/laptop/supercomputer)
- What if the attacker is quantum?

Quantum Computers

- Computers that use laws of quantum rather than classical physics: allow exponential speedups in some cases
- Most current cryptography relies on hardness of factoring, discrete log: **broken** if quantum computers are realized

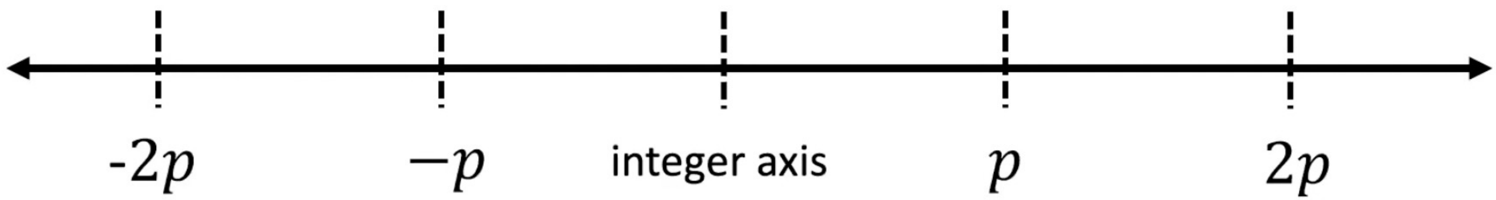


What went wrong?

- Cryptography: tightrope between structure and hardness
- Need structure for functionality, hardness for security
- RSA, DLOG: structure periodic, but carefully chosen to avoid classical efficiency, despite periodicity
- Fall prey to the “one superpower” of quantum!

Quantum Magic

- Main Idea: Cast as period finding problem
- Goal: Find p in polylog p given oracle O_p

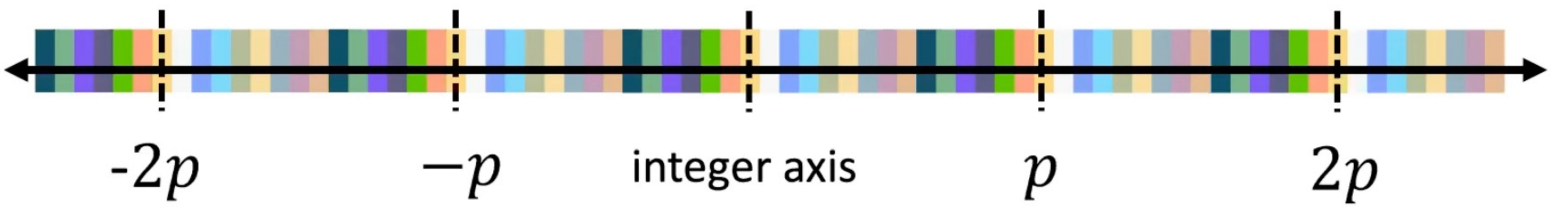


- Easy classically if $O_p : x \rightarrow x \bmod p$
- What if cosets have random names?

$O_p : x \rightarrow \text{Colour}(x \bmod p)$

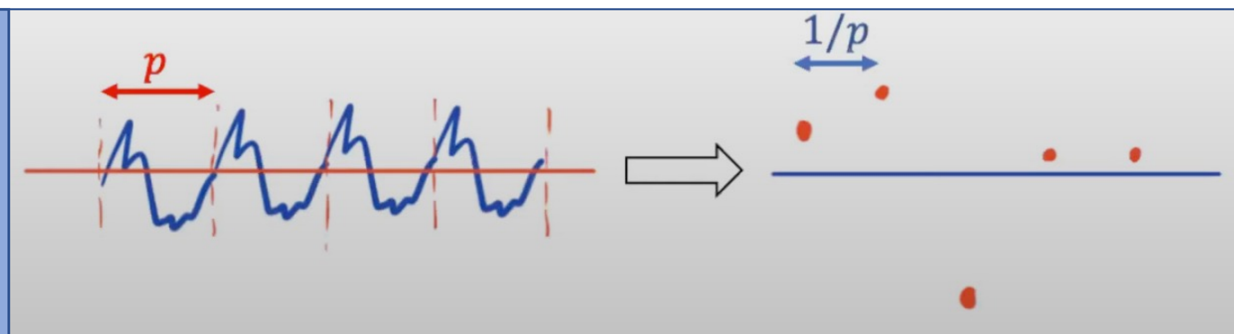
Quantum Magic

- Main Idea: Cast as period finding problem
- Goal: Find p in polylog p given oracle O_p



O_p : x → Colour (x mod p)

Quantum Poly time via Quantum Fourier Transform



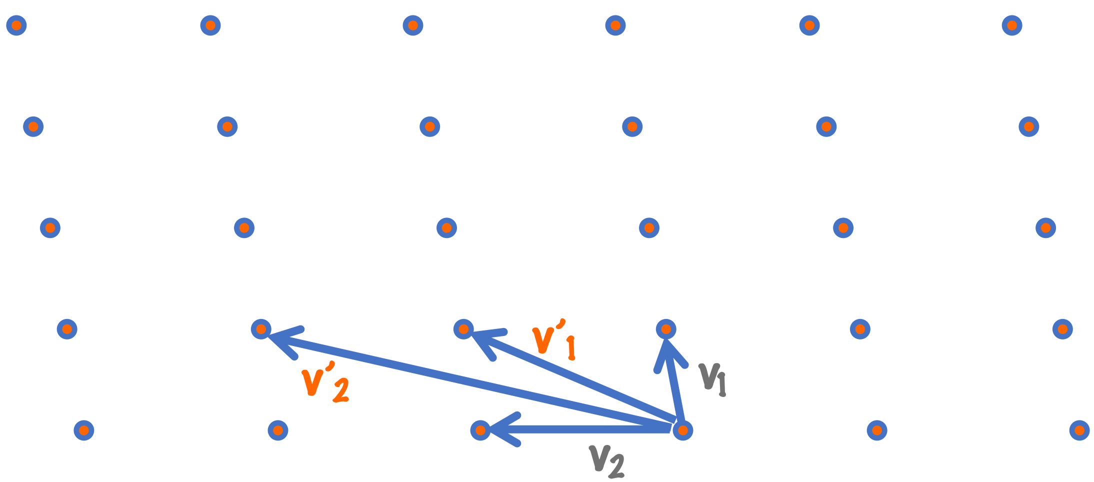
Slide Credit: Zvika Brakerski

Or does it?

Post Quantum Cryptography?

- Base hardness on mathematical problems for which quantum computers offer no advantage

- Most promising: problems in high dimensional lattices.



Cryptography from Lattices

- Post quantum secure: quantum computers do not seem to break lattice based constructions (so far)
  - Quantum algorithms do not effectively use geometry of problem
  - Need way to solve non-commutative version of HSP
- Strong security: breaking cryptosystem implies ability to solve hard problems in the worst case
- Efficient operations, parallelizable
- Enables cryptography for big data

Other Post Quantum Options

- Codes: hardness of decoding general linear codes
- Multivariate Polynomials: hardness of solving system of nonlinear multivariate polynomial equations
- Hash based: hardness of solving cryptographic hash functions
- Isogenies: based on algebraic maps between elliptic curves

NIST PQC Overview

NIST ran competition to create PQC standards

Post-Quantum Cryptography PQC

Selected Algorithms 2022

Official comments on the Selected Algorithms should be submitted using the "Submit Comment" link for the appropriate algorithm. Comments from the pqc-forum Google group subscribers will also be forwarded to the pqc-forum Google group list. We will periodically post and update the comments received to the appropriate algorithm.

All relevant comments will be posted in their entirety and should not include PII information in the body of the email message.

Please refrain from using OFFICIAL COMMENT to ask administrative questions, which should be sent to pqc-comments@nist.gov

History of Selected Algorithms Updates

Selected Algorithms: Public-key Encryption and Key-establishment Algorithms

Algorithm	Algorithm Information	Submitters	Comments
CRYSTALS-KYBER **PQC License Summary & Excerpts**	Zip File (7MB) IP Statements Website	Peter Schwabe Roberto Avanzi Joppe Bos Leo Ducas Eike Kiltz Tancrede Lepoint Vadim Lyubashevsky John M. Schanck Gregor Seiler Damien Stehle Jintai Ding	Submit Comment View Comments

Selected Algorithms: Digital Signature Algorithms

Algorithm	Algorithm Information	Submitters	Comments
CRYSTALS-DILITHIUM	Zip File (11MB) IP Statements Website	Vadim Lyubashevsky Leo Ducas Eike Kiltz Tancrede Lepoint Peter Schwabe Gregor Seiler Damien Stehle Shi Bai	Submit Comment View Comments
FALCON	Zip File (4MB) IP Statements Website	Thomas Prest Pierre-Alain Fouque Jeffrey Hoffstein Paul Kirchner Vadim Lyubashevsky Thomas Pornin Thomas Ricosset Gregor Seiler William Whyte Zhenfei Zhang	Submit Comment View Comments
SPHINCS+	Zip File (230MB) IP Statements Website	Andreas Hulsing Daniel J. Bernstein Christoph Dobraunig Maria Eichlseder Scott Fluhrer Stefan-Lukas Gazdag Panos Kampanakis Stefan Kolbl Tanja Lange Martin M Lauridsen Florian Mendel Ruben Niederhagen Christian Rechberger Joost Rijneveld Peter Schwabe Jean-Philippe Aumasson Bas Westerbaan Ward Beullens	Submit Comment View Comments

Bumpy road

Breaking Rainbow Takes a Weekend on a Laptop

Ward Beullens

IBM Research, Zurich, Switzerland
wbe@zurich.ibm.com

Abstract. This work introduces new key recovery attacks against the Rainbow signature scheme, which is one of the three finalist signature schemes still in the NIST Post-Quantum Cryptography standardization project. The new attacks outperform previously known attacks for all the parameter sets submitted to NIST and make a key-recovery practical for the SL 1 parameters. Concretely, given a Rainbow public key for the SL 1 parameters of the second-round submission, our attack returns the corresponding secret key after on average 53 hours (one weekend) of computation time on a standard laptop.

Many ups and downs

AN EFFICIENT KEY RECOVERY ATTACK ON SIDH (PRELIMINARY VERSION)

WOUTER CASTRYCK AND THOMAS DECRU

imec-COSIC, KU Leuven

Abstract. We present an efficient key recovery attack on the Supersingular Isogeny Diffie–Hellman protocol (SIDH), based on a "glue-and-split" theorem due to Kani. Our attack exploits the existence of a small non-scalar endomorphism on the starting curve, and it also relies on the auxiliary torsion point information that Alice and Bob share during the protocol. Our Magma implementation breaks the instantiation `SIKEp434`, which aims at security level 1 of the Post-Quantum Cryptography standardization process currently ran by NIST, in about one hour on a single core. This is a preliminary version of a longer article in preparation.

Still unclear which to use?

Bottomline: cannot ignore the math!

Slide Credit: Damien Stehle

Exciting New Applications

Encrypted Computation
Personalised Medicine

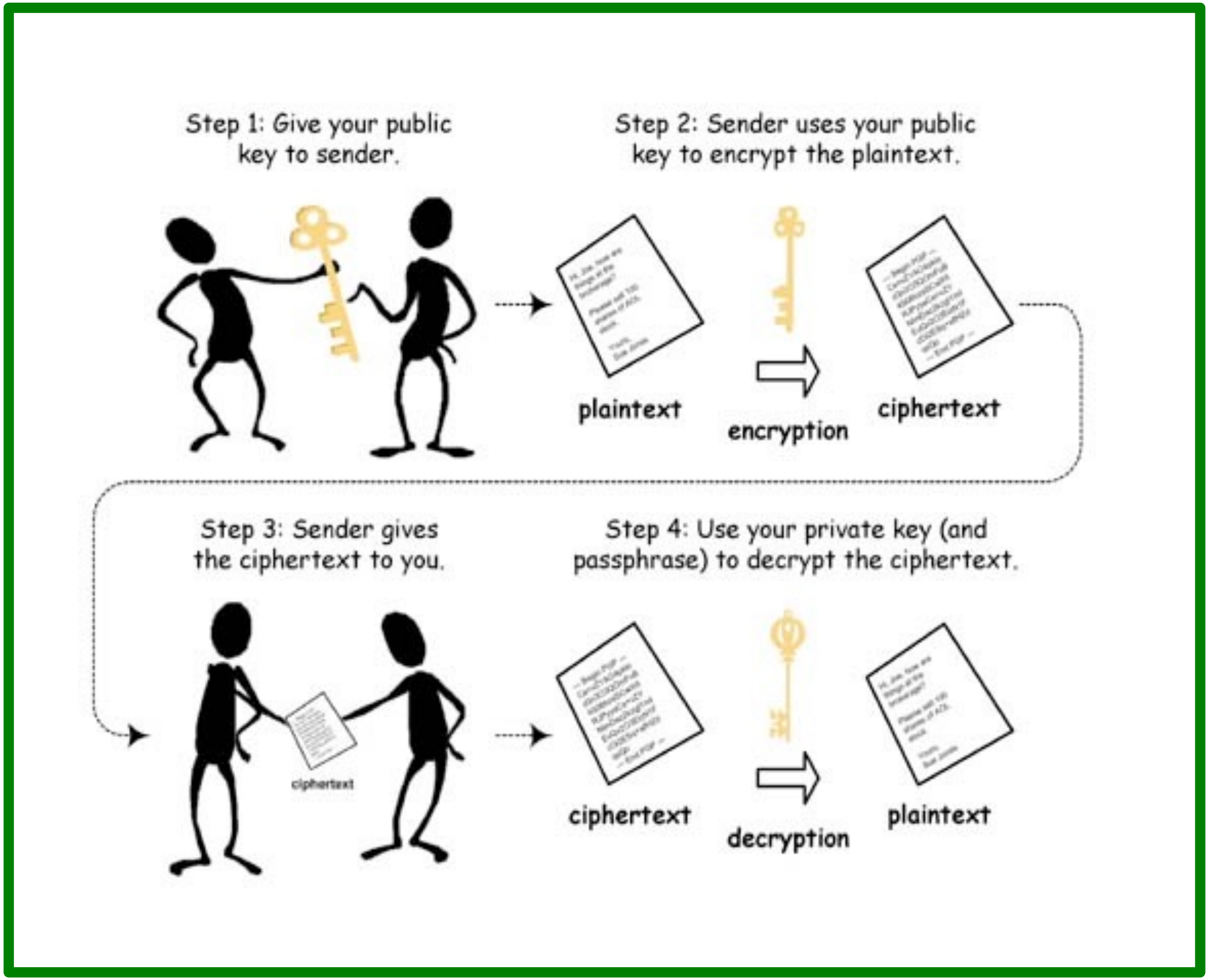
"The dream for tomorrow's medicine is to understand the links between DNA and disease — and to tailor therapies accordingly. But scientists have a problem: how to keep genetic data and medical records secure while still enabling the massive, cloud-based analyses needed to make meaningful associations."

"You don't look anything like the long haired, skinny kid I married 25 years ago. I need a DNA sample to make sure it's still you."

Check Hayden, E. (2015). *Nature*, *519*, 400-401.

Can Cryptography solve this?

Public Key Encryption



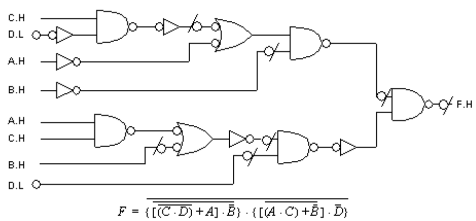
PKE does not suffice!

- Secret keys correspond to users
- Encrypt for each user?
- All or nothing access
  - Genomic data (for instance) is too sensitive to share
  - May be willing to participate in study which reveals output (result of study) without revealing input (personal data)

More Expressive Encryption

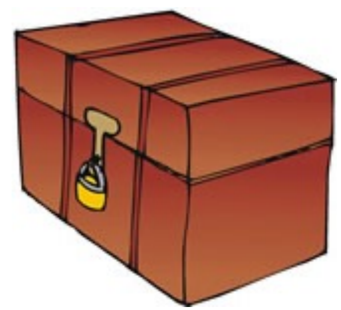
Functional Encryption!

Secret Keys
for functions *F*



+

Ciphertexts
for inputs *x*



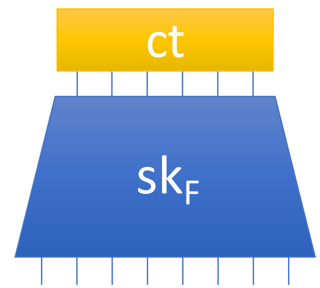
Decryption recovers F(x)

F : Age distribution of people with lung cancer
X : particular user's disease profile

Encryption with Partial Decryption Keys

Encrypt (x):

Decrypt (sk_F, ct):



$y = F(x)$

Keygen(F):

Security:
Adversary possessing keys for multiple circuits F_i cannot distinguish $Enc(x_0)$ from $Enc(x_1)$ unless $F_i(x_0) \neq F_i(x_1)$

Functional Encryption [SW05,BSW11]

Personalized Medicine?

Encrypt
input = genomic data of users

ct(Deepro)

ct(Supriyo)

ct(Kunal)

ct(Anuja)

Decrypt (sk_F, ct):

ct(Deepro) ct(Supriyo)

ct(Kunal) ct(Anuja)

y = F(x)

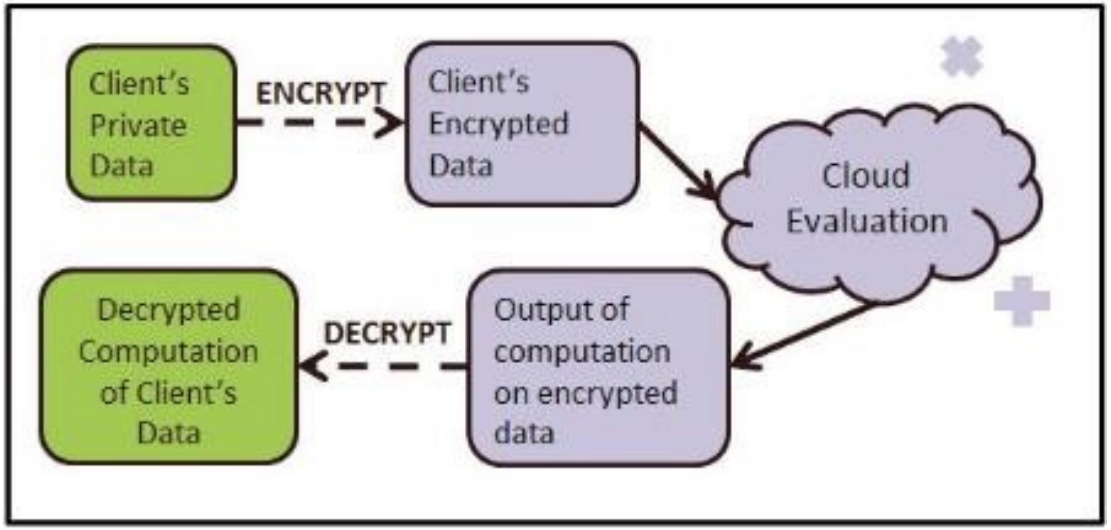
Keygen
input: some medical research algo

Security: No one's personal genomic data is leaked!

Functional Encryption [SW05,BSW11]

Fully Homomorphic Encryption

[G09, BV11, BGV12, GSW13…]



Expressive Functionality: Supports arbitrary circuits

Compact ciphertext, independent of circuit size

Encryption and function evaluation commute!
Enc(f(x)) =* f(Enc(x))

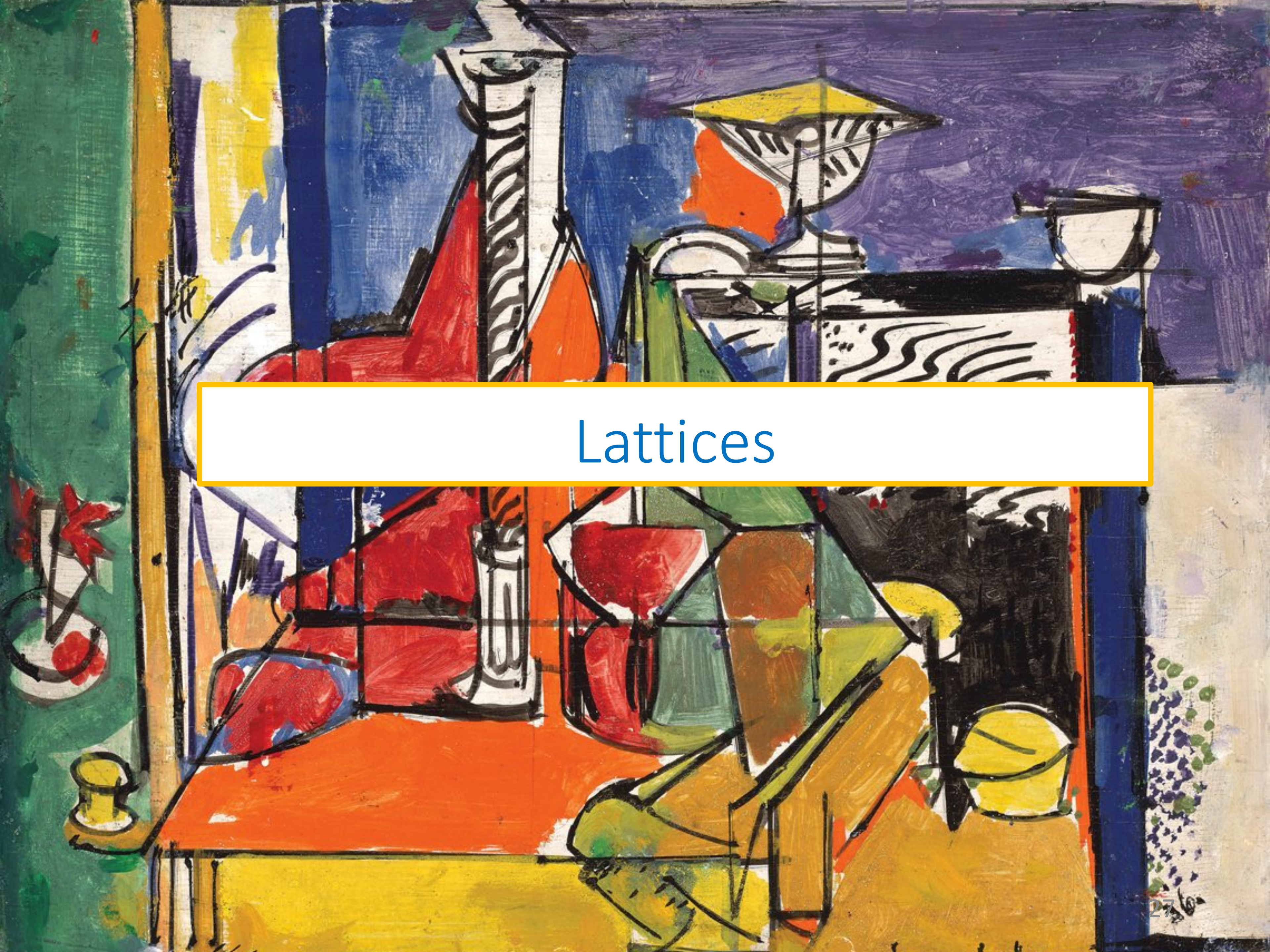
* : roughly

Cryptography from Lattices

- Redo old cryptography:
  - build post-quantum versions of existing functionalities
- Build new functionalities
  - not realizable before

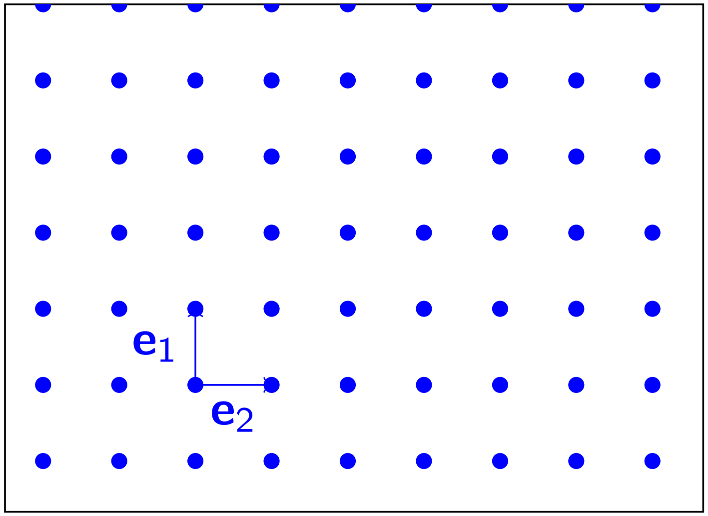
Caveat: currently at cost of efficiency

In Crypto-land, its always party-time!



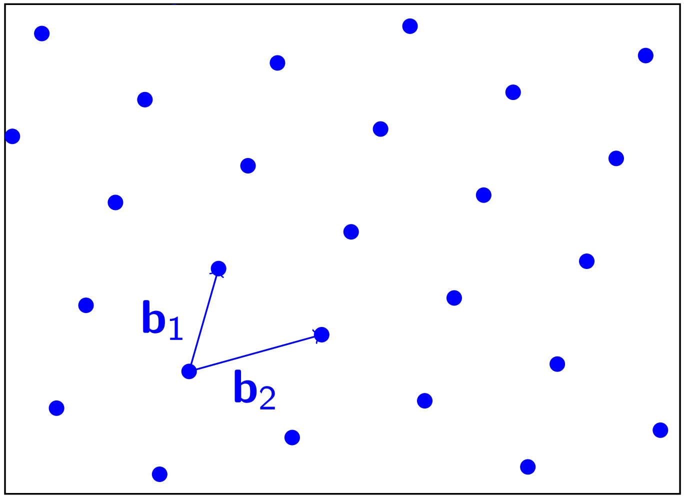
Lattices

What is a lattice?



The simplest lattice in n-dimensional space is the integer lattice

$$\Lambda = \mathbb{Z}^n$$



Other lattices are obtained by applying a linear transformation

$$\Lambda = \mathbf{B}\mathbb{Z}^n \qquad (\mathbf{B} \in \mathbb{R}^{d\times n})$$

A set of points with periodic arrangement

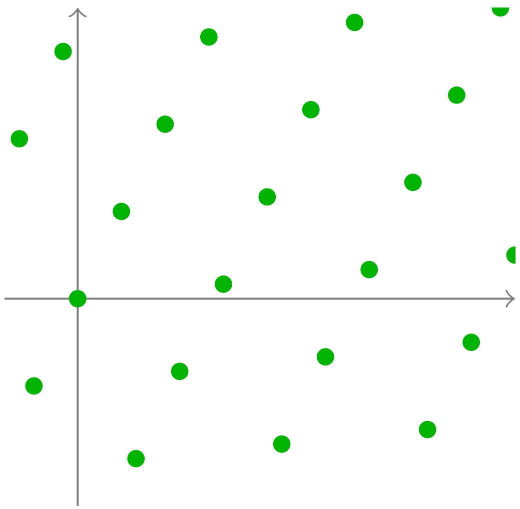
Lattices and Bases

A lattice is the set of all integer linear combinations of (linearly independent) basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$:

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{b}_i \cdot \mathbb{Z} = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{Z}^n\}$$

The same lattice has many bases

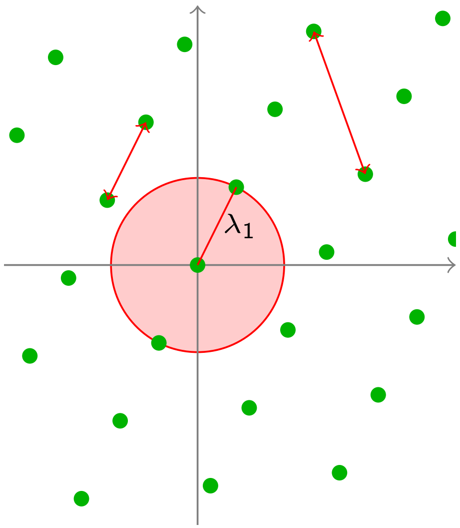
$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{c}_i \cdot \mathbb{Z}$$



Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$



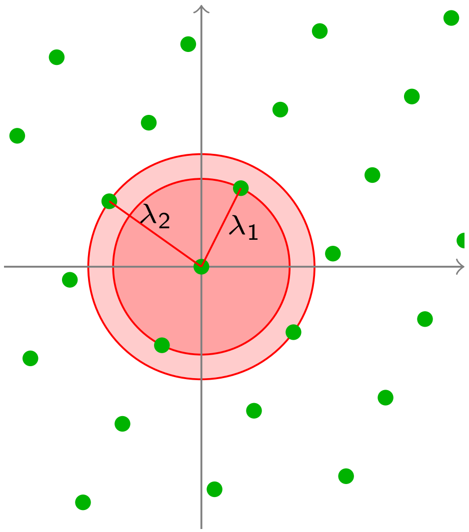
- Successive minima $(i = 1, \ldots, n)$

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

Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$



- Successive minima $(i = 1, \ldots, n)$

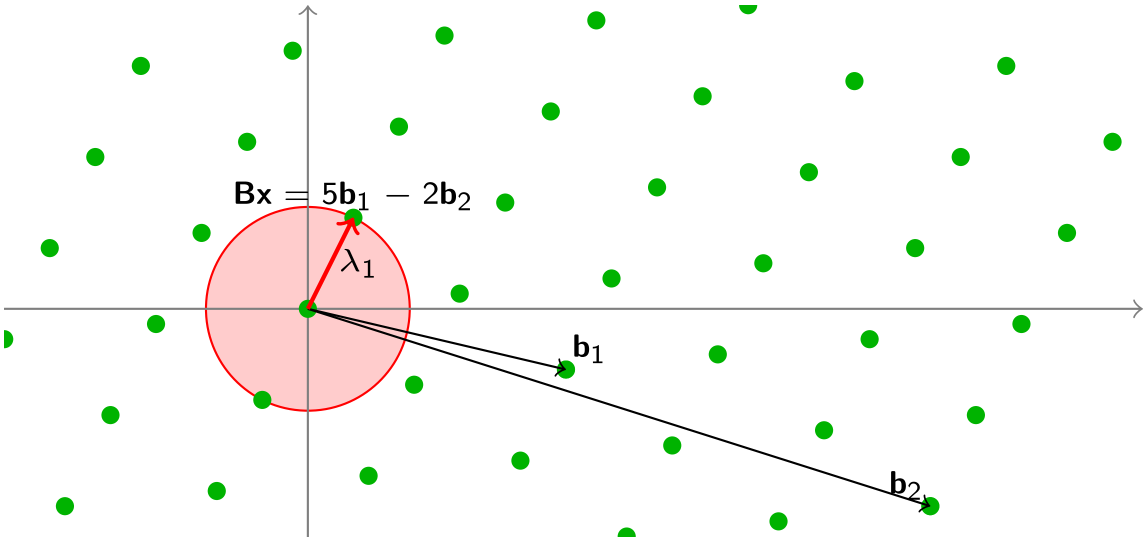
$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$

- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

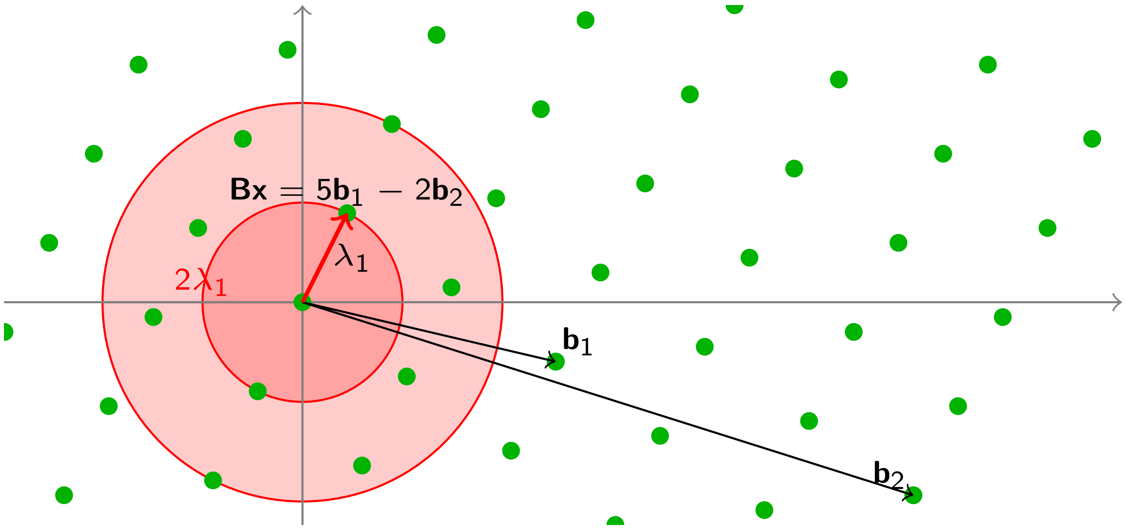
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Approximate Shortest Vector Problem

Definition (Shortest Vector Problem, SVP_γ)

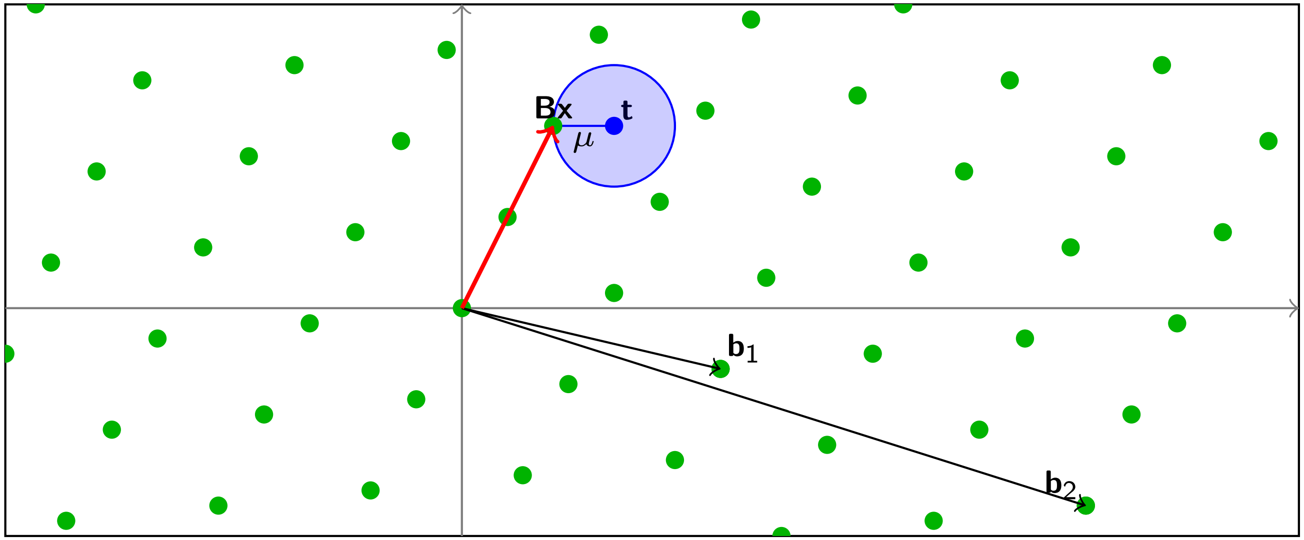
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \gamma\lambda_1$



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

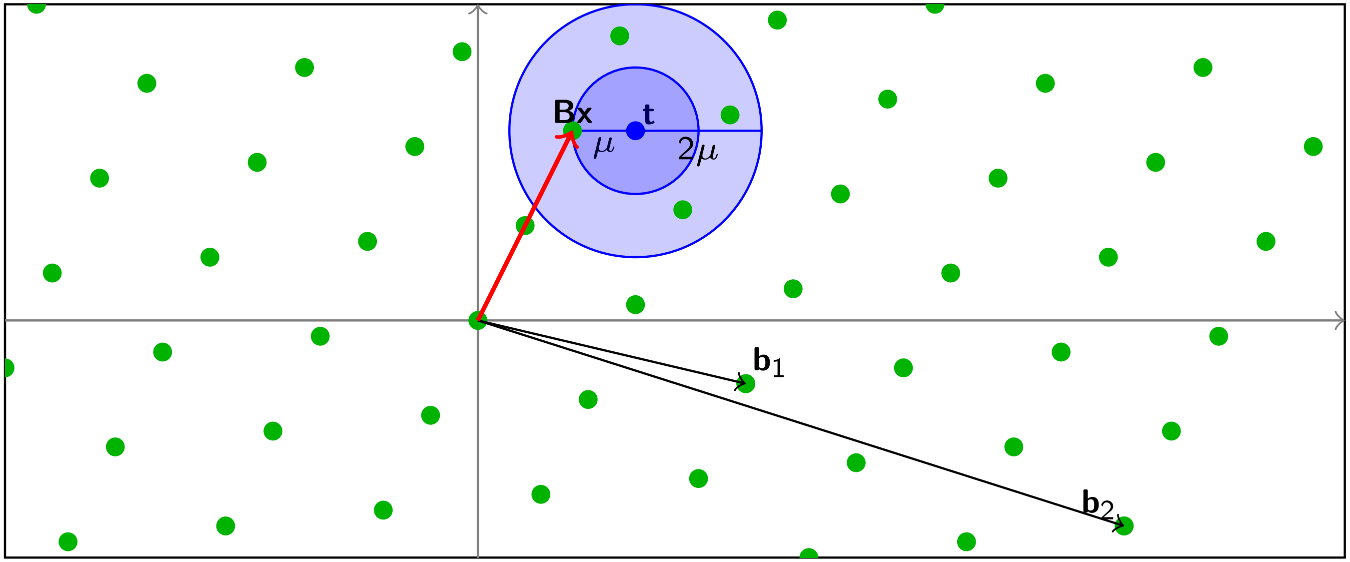
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \mu$ from the target



Approximate Closest Vector Problem

Definition (Closest Vector Problem, CVP_γ)

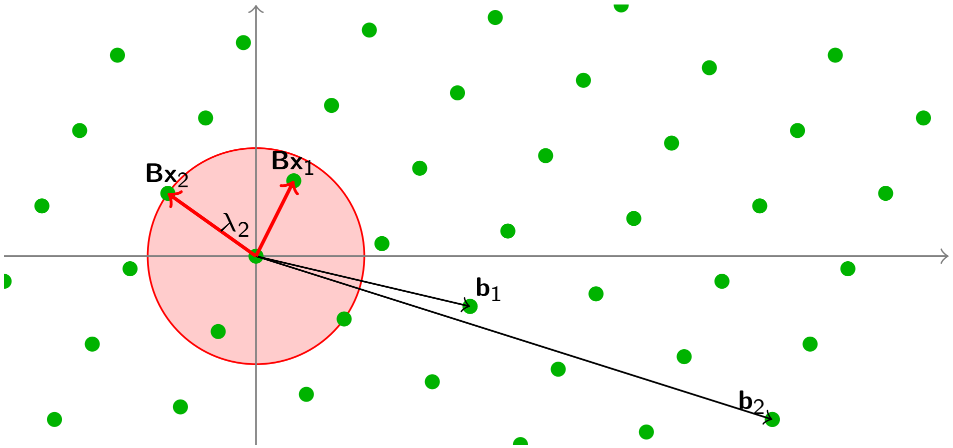
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \gamma\mu$ from the target



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

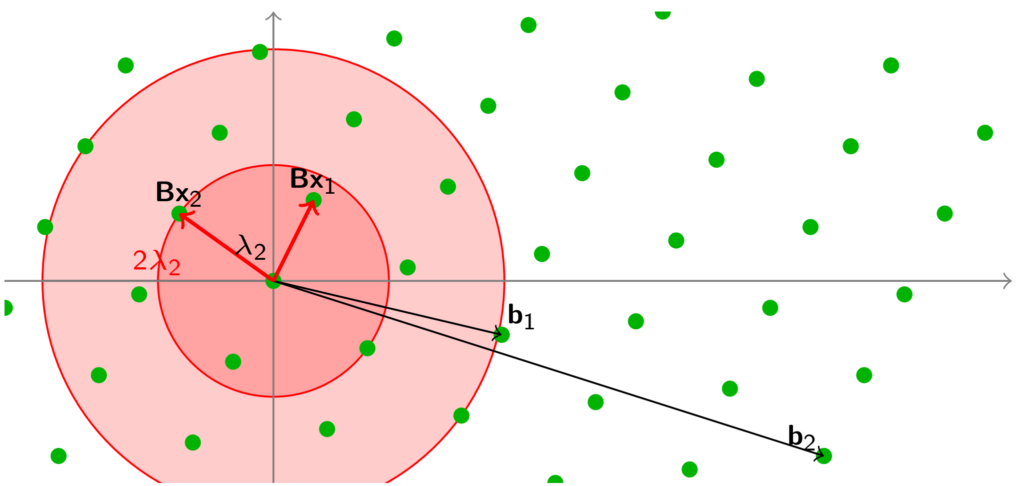
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



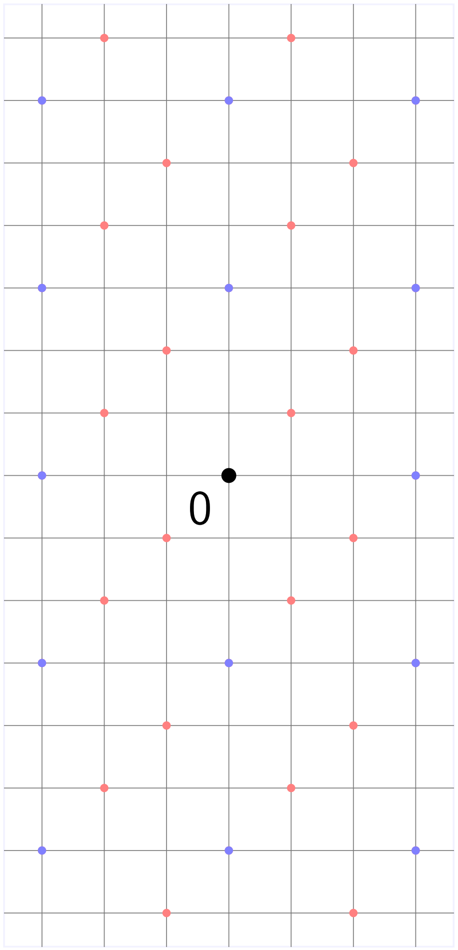
Approximate Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP_γ)

Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \gamma\lambda_n$

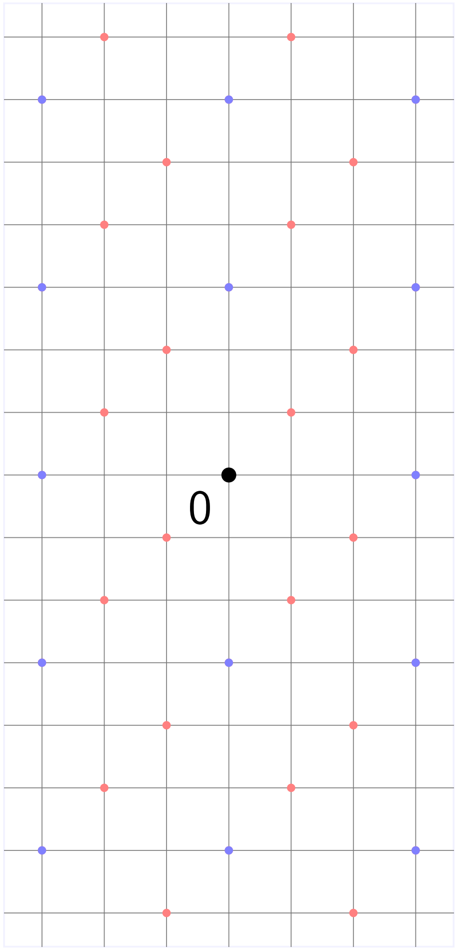


Random Lattices in Cryptography



- Cryptography typically uses (random) lattices Λ such that
  - $\Lambda \subseteq \mathbb{Z}^d$ is an integer lattice
  - $q\mathbb{Z}^d \subseteq \Lambda$ is periodic modulo a small integer q.
- Cryptographic functions based on q-ary lattices involve only arithmetic modulo q.

Random Lattices in Cryptography

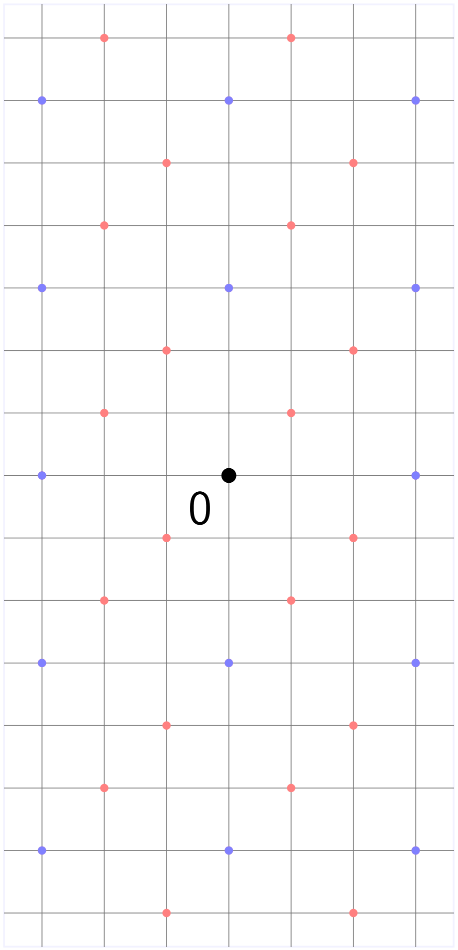


- Cryptography typically uses (random) lattices Λ such that
  - $\Lambda \subseteq \mathbb{Z}^d$ is an integer lattice
  - $q\mathbb{Z}^d \subseteq \Lambda$ is periodic modulo a small integer q.
- Cryptographic functions based on q-ary lattices involve only arithmetic modulo q.

Definition (q-ary lattice)

Λ is a q-ary lattice if $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$

Random Lattices in Cryptography



- Cryptography typically uses (random) lattices Λ such that
  - $\Lambda \subseteq \mathbb{Z}^d$ is an integer lattice
  - $q\mathbb{Z}^d \subseteq \Lambda$ is periodic modulo a small integer q.
- Cryptographic functions based on q-ary lattices involve only arithmetic modulo q.

Definition (q-ary lattice)

Λ is a q-ary lattice if $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$

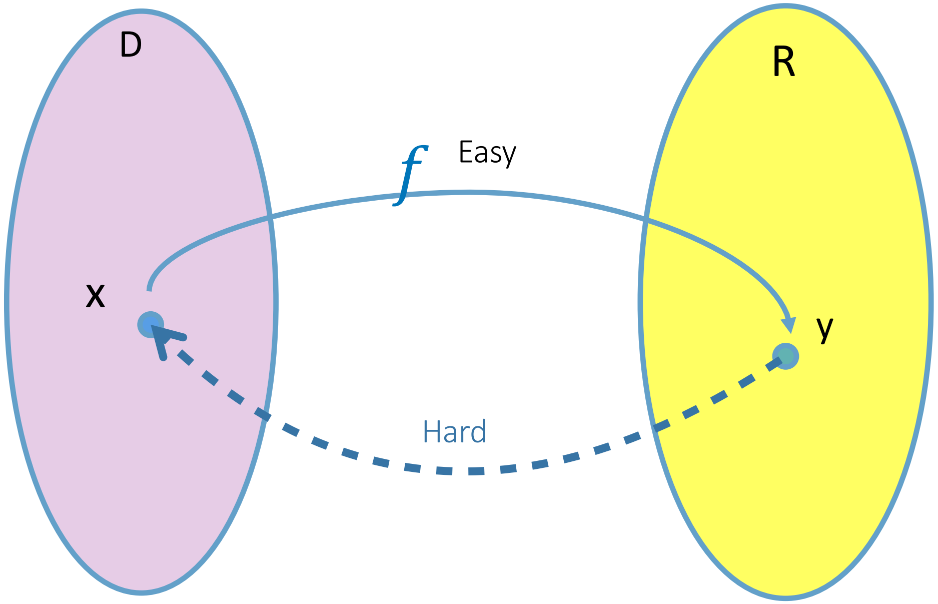
Examples (for any $\mathbf{A} \in \mathbb{Z}_q^{n \times d}$)

- $\Lambda_q(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{x} \bmod q \in \mathbf{A}^T\mathbb{Z}_q^n\} \subseteq \mathbb{Z}^d$
- $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\} \subseteq \mathbb{Z}^d$

Building Cryptography

One Way Functions

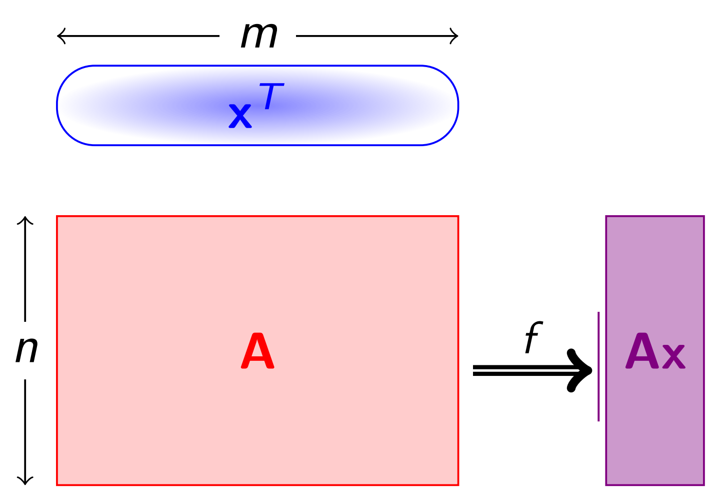
$f: D \to R$, **One Way**



Most basic "primitive" in cryptography!

Ajtai's One Way Function

- Parameters: $m, n, q \in \mathbb{Z}$
- Key: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$
- Input: $\mathbf{x} \in \{0,1\}^m$
- Output: $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$

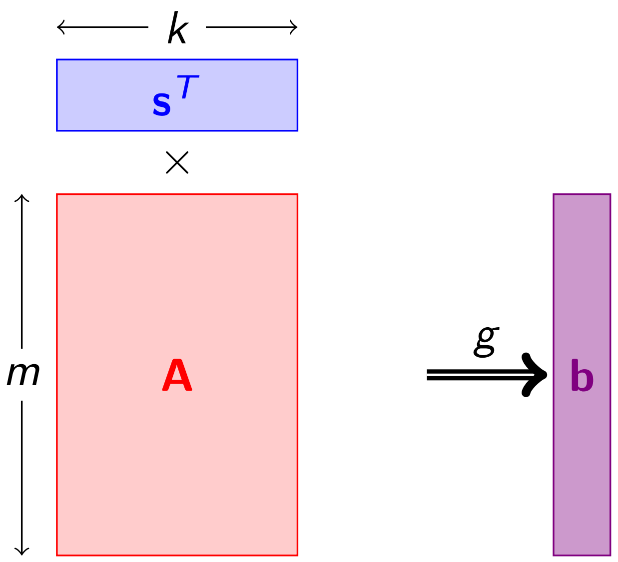


Theorem (A'96)

For $m > n \lg q$, if lattice problems (SIVP) are hard to approximate in the worst-case, then $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$ is a one-way function.

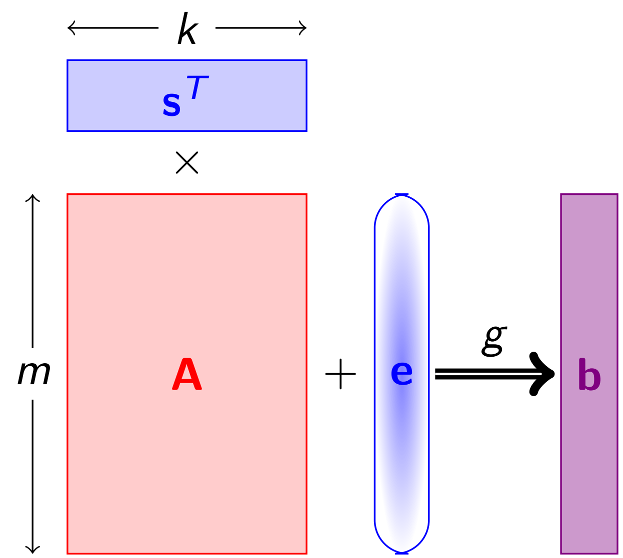
Regev's One Way Function

- $\mathbf{A} \in \mathbb{Z}_q^{m \times k}$, $\mathbf{s} \in \mathbb{Z}_q^k$, $\mathbf{e} \in \mathcal{E}^m$.
- $g_{\mathbf{A}}(\mathbf{s} \quad) = \mathbf{A}\mathbf{s} \qquad \bmod q$



Regev's One Way Function

- $\mathbf{A} \in \mathbb{Z}_q^{m \times k}$, $\mathbf{s} \in \mathbb{Z}_q^k$, $\mathbf{e} \in \mathcal{E}^m$.
- $g_{\mathbf{A}}(\mathbf{s}; \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$
- Learning with Errors: Given $\mathbf{A}$ and $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$, recover $\mathbf{s}$.



Theorem (R'05)

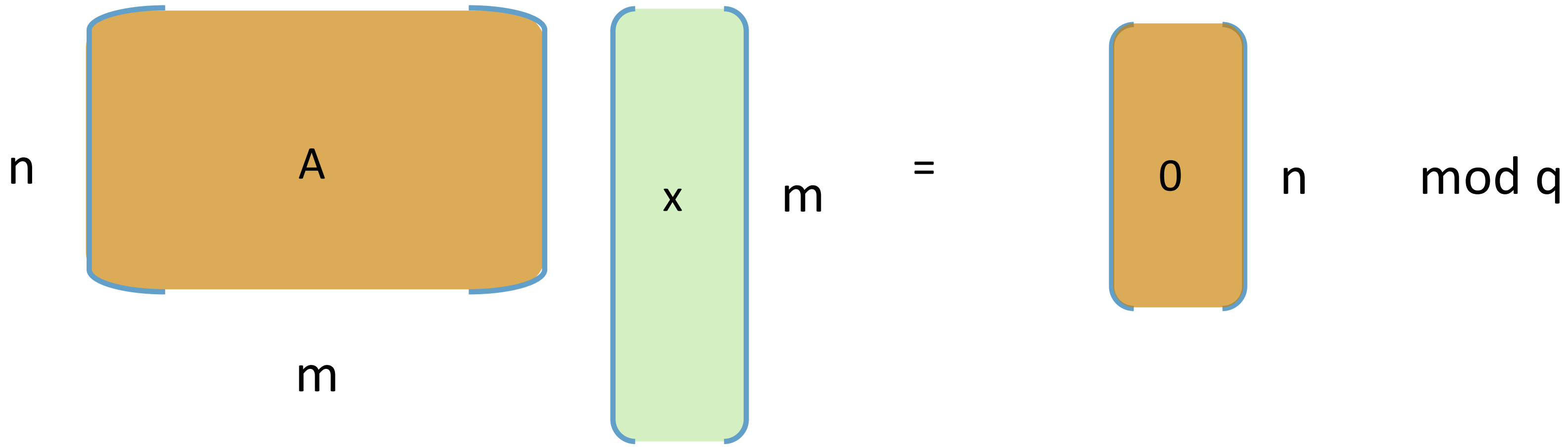
The function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ is hard to invert on the average, assuming SIVP is hard to approximate in the worst-case.

Short Integer Solution Problem

Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, q = \text{poly}(n), m = \Omega(n \log q)$

Given matrix **A**, find "short" (low norm) vector **x** such that

$$\mathbf{Ax} = 0 \;\; mod \;\; q \in \mathbb{Z}_q^n$$



Learning With Errors Problem

Distinguish "noisy inner products" from uniform

Fix uniform $s \in Z_q^n$

a_1 , $b_1 = \langle a_1,s\rangle + e_1$ a_2 , $b_2 = \langle a_2,s\rangle + e_2$ ⋮ a_m , $b_m = \langle a_m,s\rangle + e_m$	vs	a'_1 , b'_1 a'_2 , b'_2 ⋮ a'_m , b'_m
a_i uniform $\in Z_q^n$, $e_i \sim \phi \in Z_q$		a_i uniform $\in Z_q^n$, b_i uniform $\in Z_q$

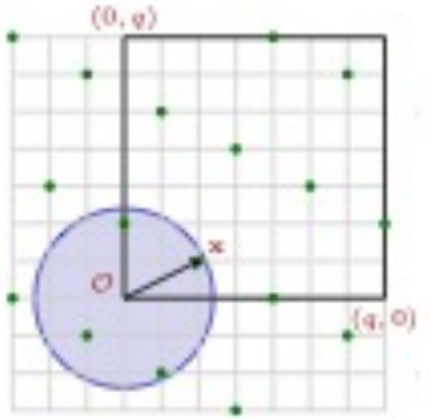
Recap:Lattice Based One Way Functions

$$Public\ Key\ \mathbf{A} \in \mathbb{Z}_q^{n \times m}, q = \text{poly}(n), m = \Omega(n\log q)$$

Based on SIS

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x}\ mod\ q \in \mathbb{Z}_q^n$$

- Short x, surjective
- CRHF if SIS is hard



Based on LWE

$$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t mod q \in \mathbb{Z}_q^m$$

- Very short e, injective
- OWF if LWE is hard [Reg05…]

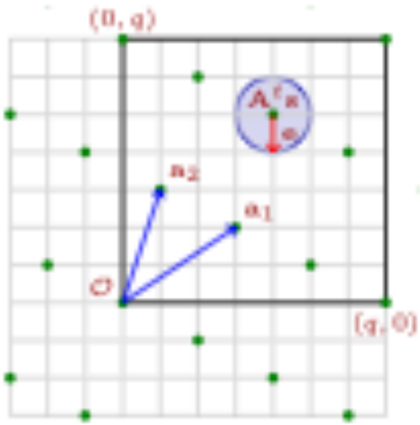
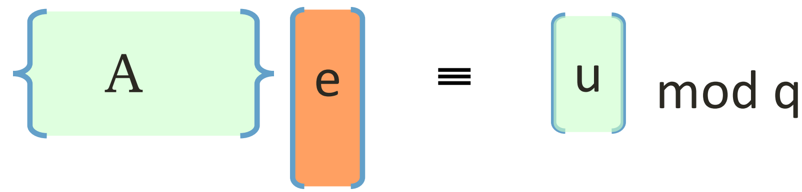


Image Credit: MP12 slides

Public Key Encryption [Regev05]

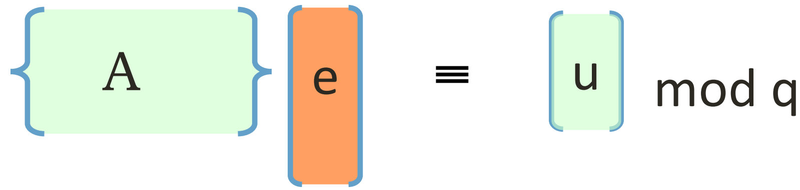
- Recall A (e) = u mod q hard to invert

- Secret: e, Public : A, u



Public Key Encryption [Regev05]

- Recall A (e) = u mod q hard to invert
- Secret: e, Public : A, u



- Encrypt (A, u) :
  - Pick random vector s
  - $c_0 = A^T s + \text{noise}$
  - $c_1 = u^T s + \text{noise} + \text{msg}$

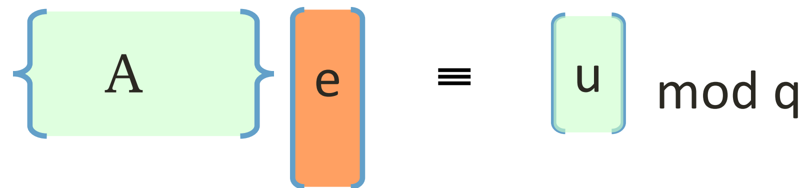
Public Key Encryption [Regev05]

- Recall $A\,(e) = u \bmod q$ hard to invert
- Secret: e, Public : A, u $\quad A \; e \equiv u \bmod q$
- Encrypt (A, u) :
  - Pick random vector s
  - $c_0 = A^T s + \text{noise}$
  - $c_1 = u^T s + \text{noise} + \text{msg}$
- Decrypt (e) :
  - $e^T c_0 - c_1 = \text{msg} + \text{noise}$

Small only if e is small

Public Key Encryption [Regev05]

- Recall A (e) = u mod q hard to invert, easy with trapdoor
- Secret: e, Public : A, u



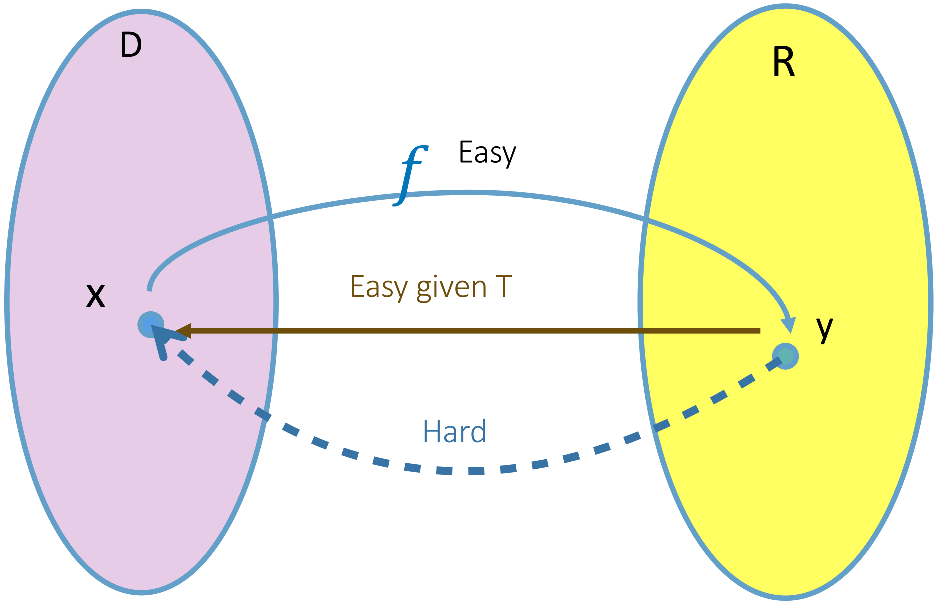
- By SIS problem, hard to find short e
- By LWE problem, ciphertext appears random
  - $c_0 = A^T s$ + noise, looks like random
  - $c_1 = u^T s$ + noise + msg , looks like random + msg
  - Hence hides message “msg”

For Signatures, need Lattice Trapdoors

Trapdoor Functions

Generate (f, T)

$f: D \rightarrow R$, One Way



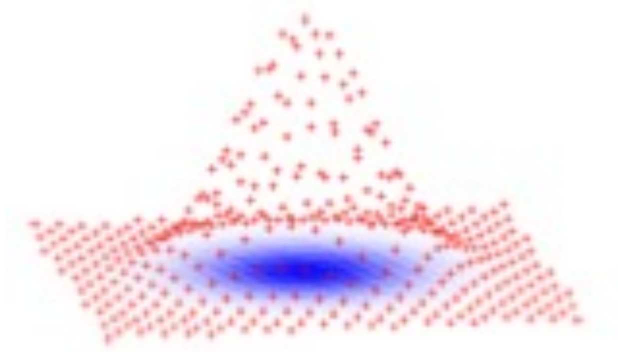
We will construct trapdoor functions from two lattice problems

Inverting functions for Crypto

- Given $\mathbf{u} = f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x}\ mod q$
- Sample

$$\mathbf{x}' \leftarrow= f_{\mathbf{A}}^{-1}(\mathbf{u})$$

with prob $\propto \exp(-\| \mathbf{x}' \|^2/\sigma^2)$

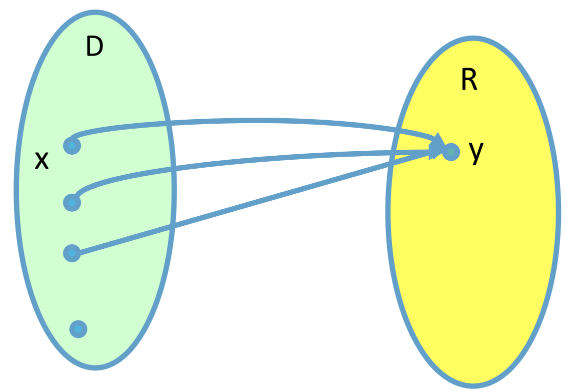


- (… $nodq$
- F

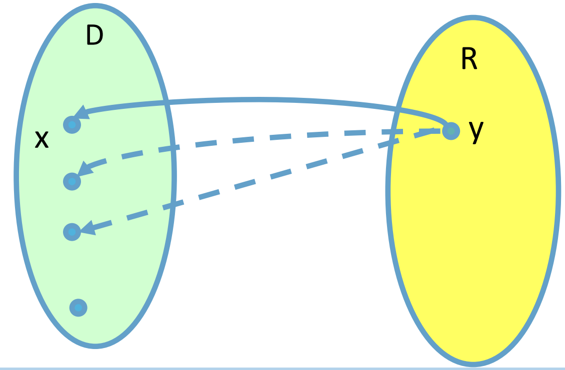
Latter distribution needs lattice trapdoors!

Preimage Sampleable Trapdoor Functions!

Generate (x, y) in two equivalent ways

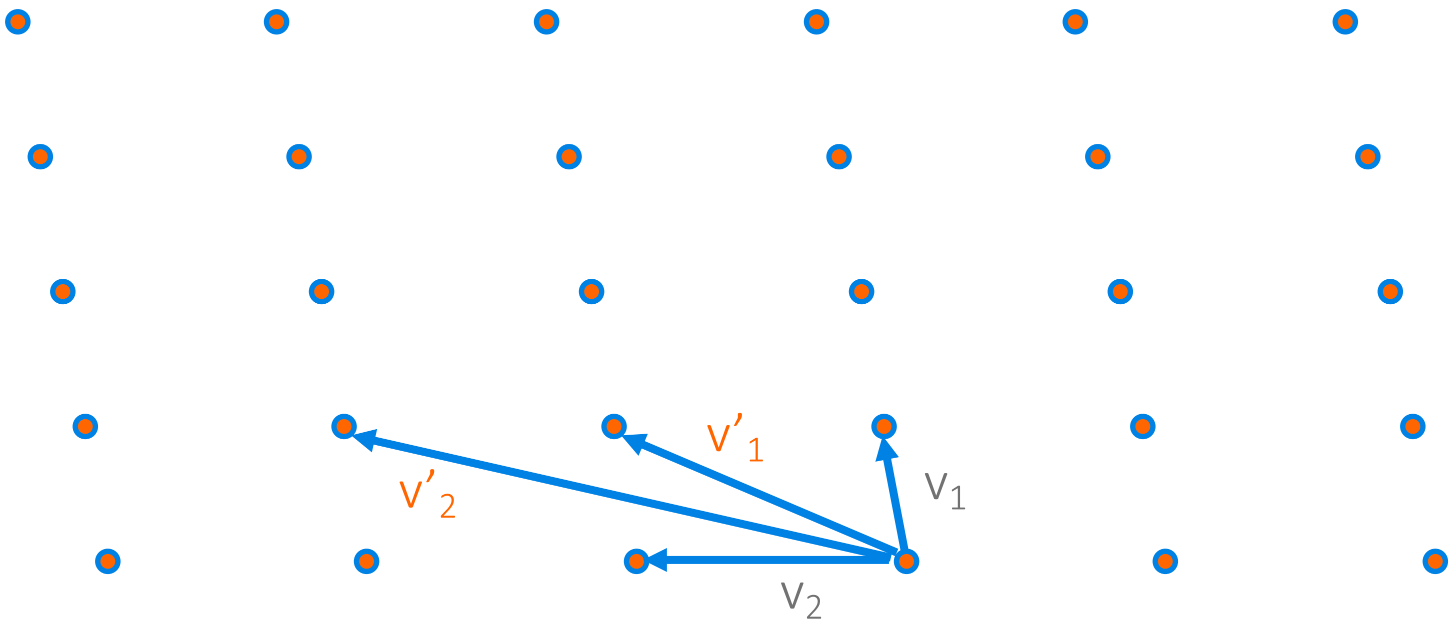


OR



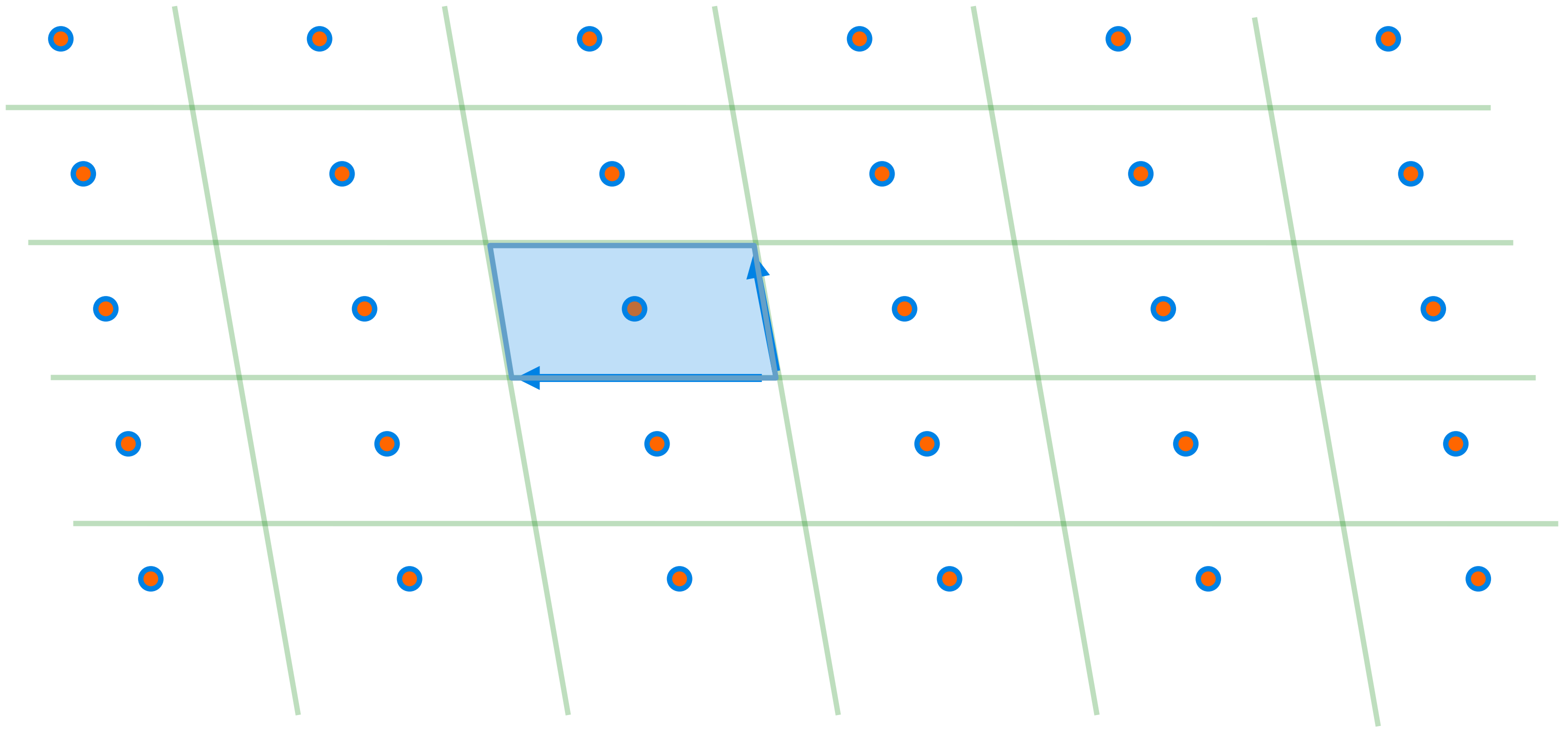
Same Distribution (Discrete Gaussian, Uniform) !

Lattice Trapdoors: Geometric View

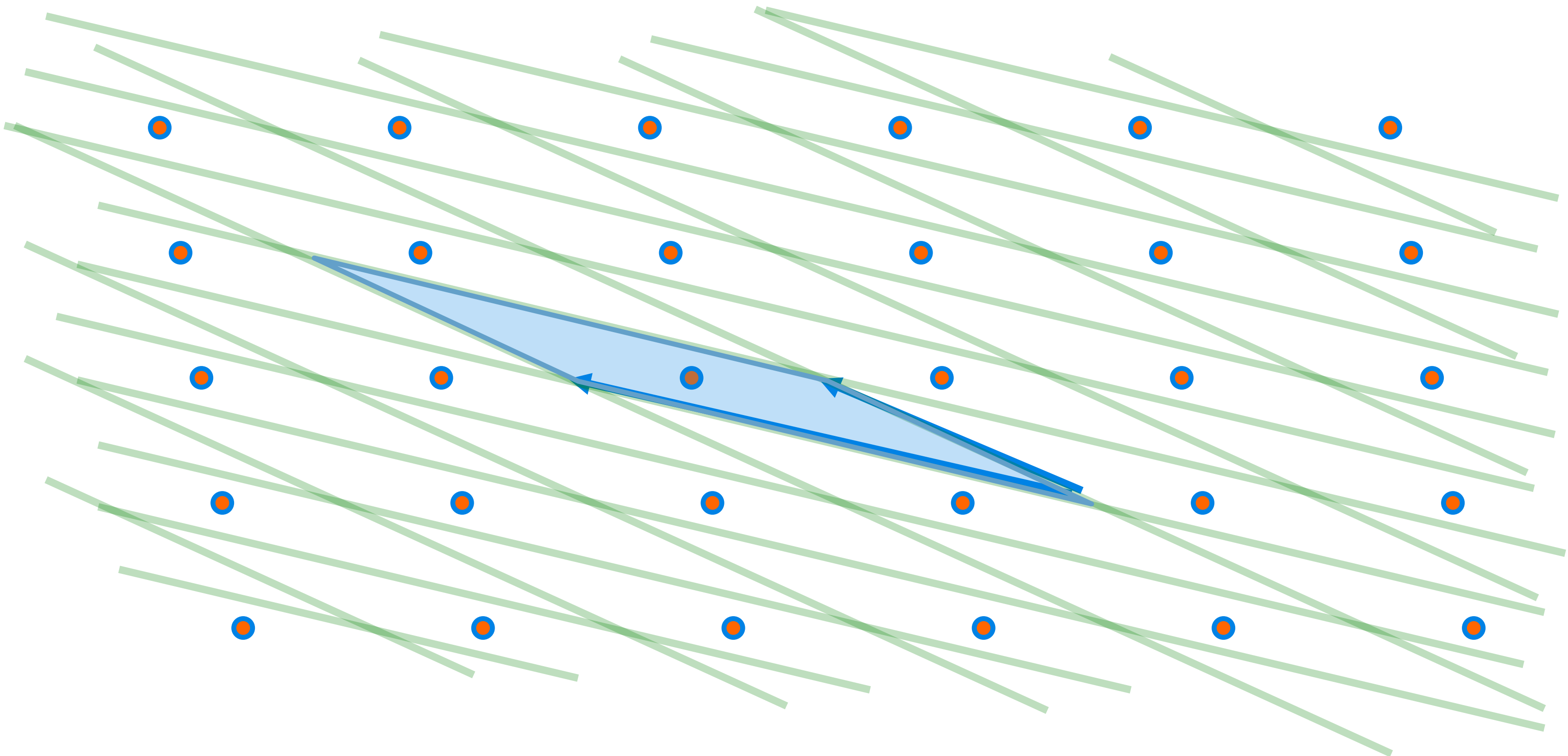


Multiple Bases

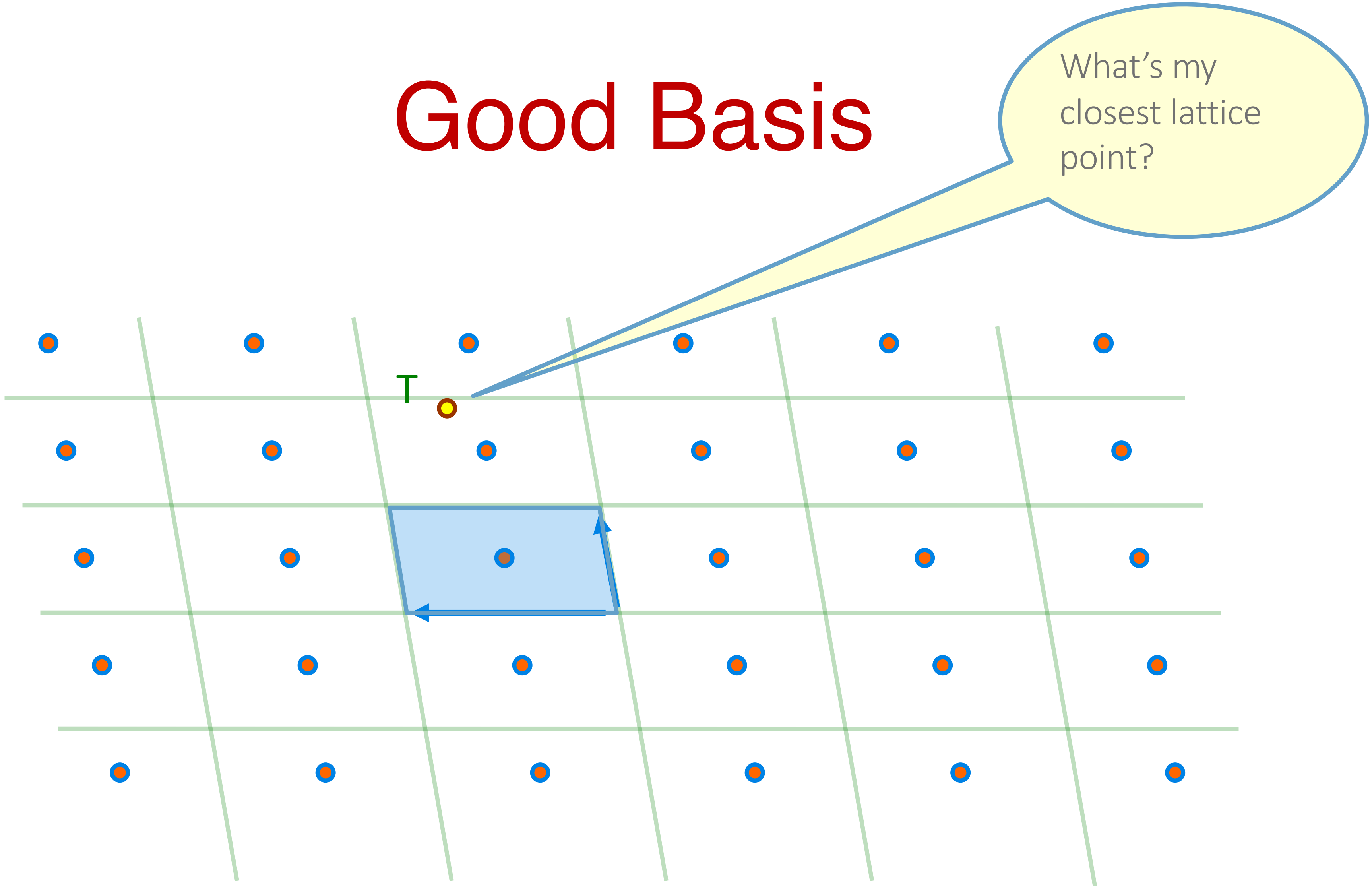
Parallelopipeds



Parallelopipeds

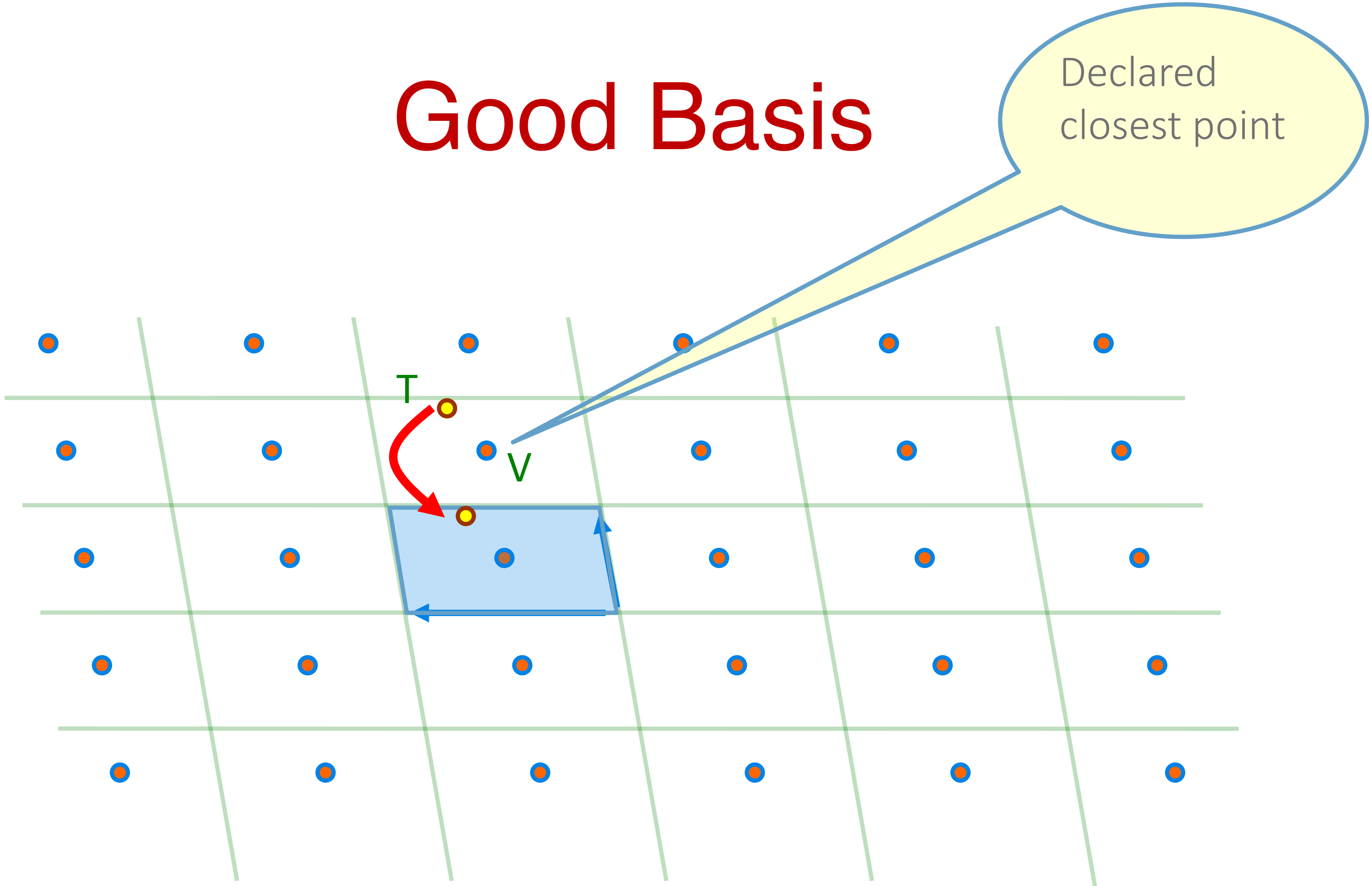


Good Basis



"Quite short" and "nearly orthogonal"

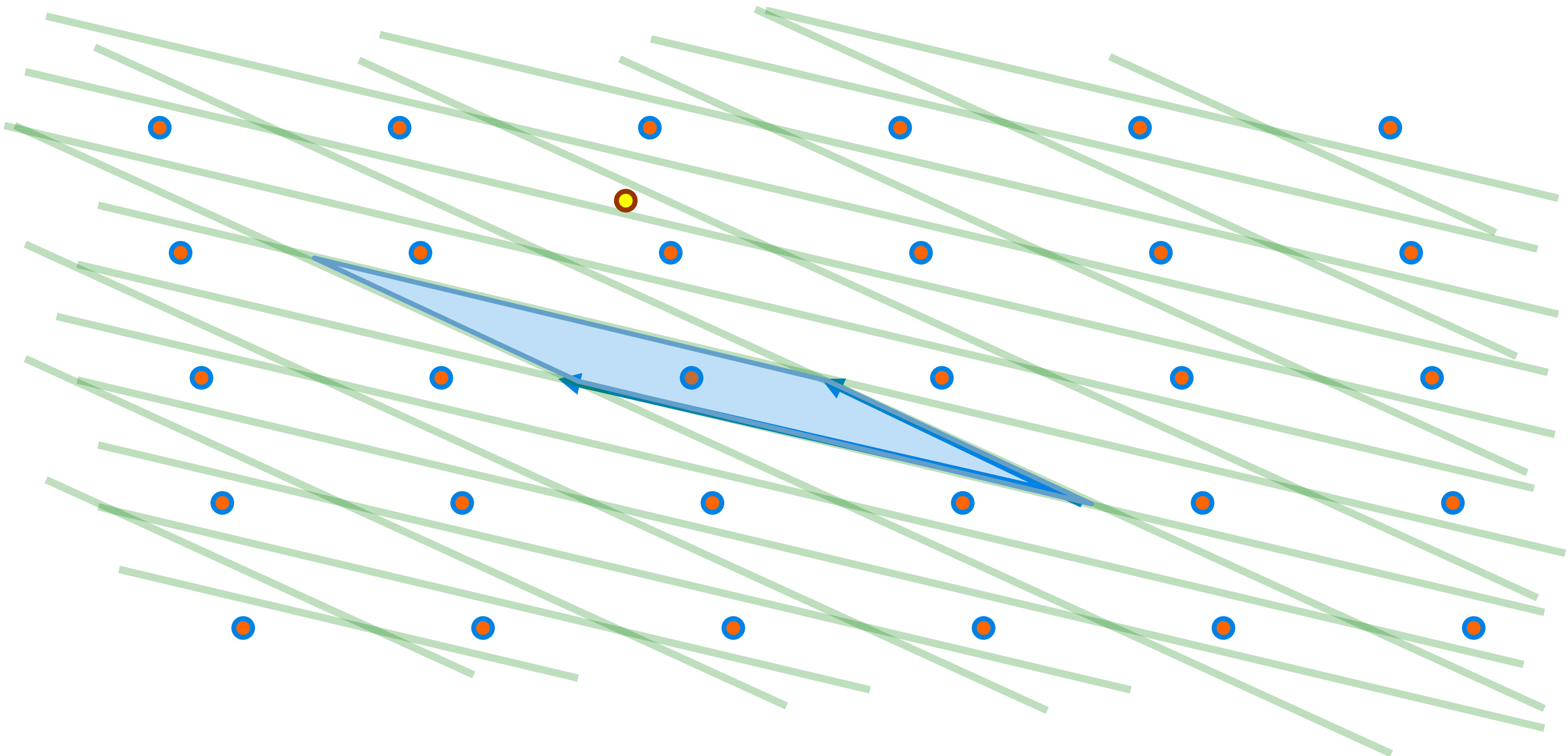
Good Basis



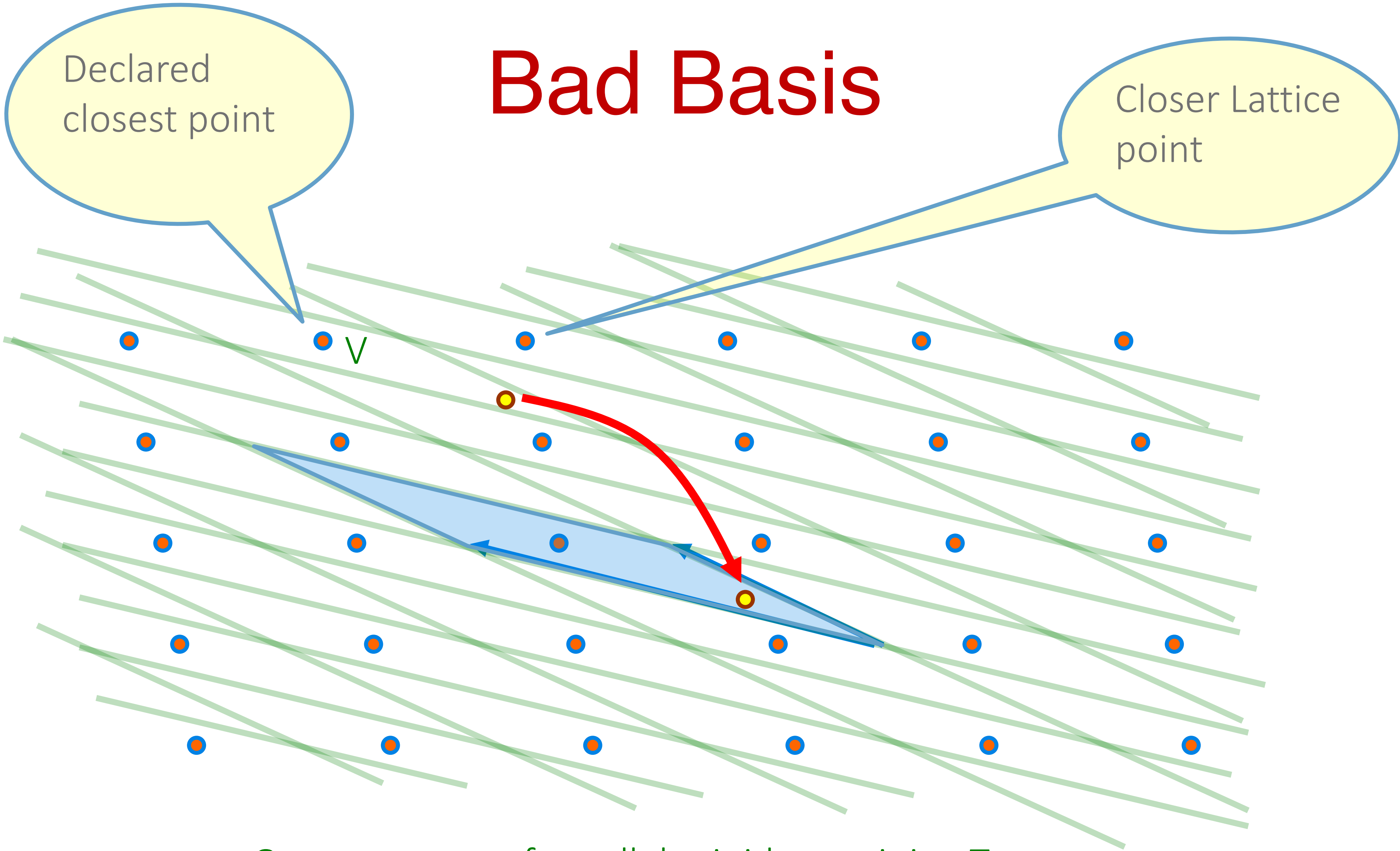
Output center of parallelopipid containing T

Pretty Accurate…

Bad Basis



Bad Basis



Output center of parallelopipid containing T

Not So Accurate…

Basis quality and Hardness

- SVP, CVP, SIS (...) hard given arbitrary (bad) basis
- Some hard lattice problems are easy given a good basis
- Will exploit this asymmetry

Use Short Basis as Cryptographic Trapdoor!

Lattice Trapdoors

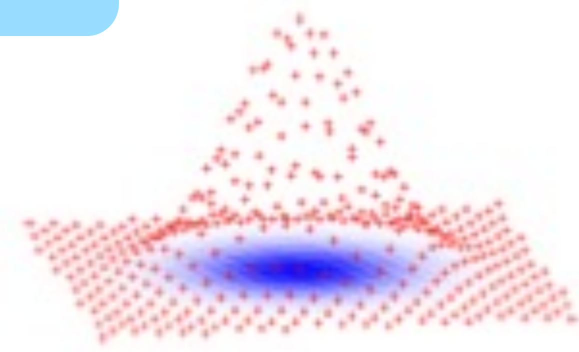
Inverting Our Function

Recall $\mathbf{u} = f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\,\mathbf{x}\, mod q$

Want

$$\mathbf{x}' \leftarrow= f_{\mathbf{A}}^{-1}(\mathbf{u})$$

with prob $\propto \exp(-\| \mathbf{x}' \|^2/\sigma^2)$

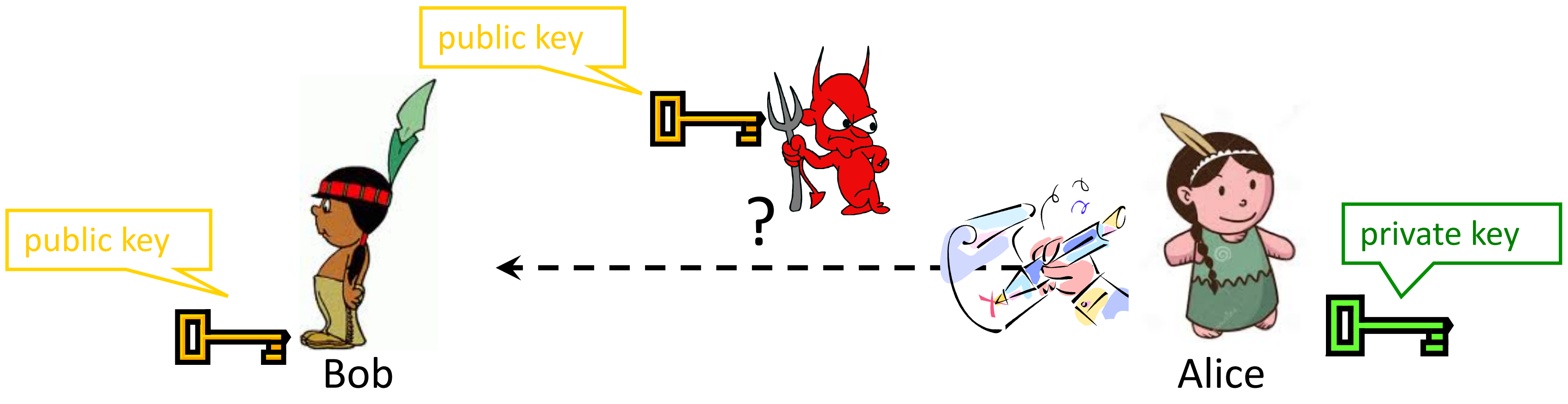


The Lattice

$$\mathbf{\Lambda} = \{\mathbf{x}: \mathbf{Ax} = 0 \; mod \; q\} \subseteq \mathbb{Z}_q^m$$

Short basis for $\mathbf{\Lambda}$ lets us sample from $f_{\mathbf{A}}^{-1}(\mathbf{u})$
with correct distribution!

Digital Signatures



Everybody knows Alice's public key
Only Alice knows the corresponding private key

<u>Goal</u>: Alice sends a "digitally signed" message

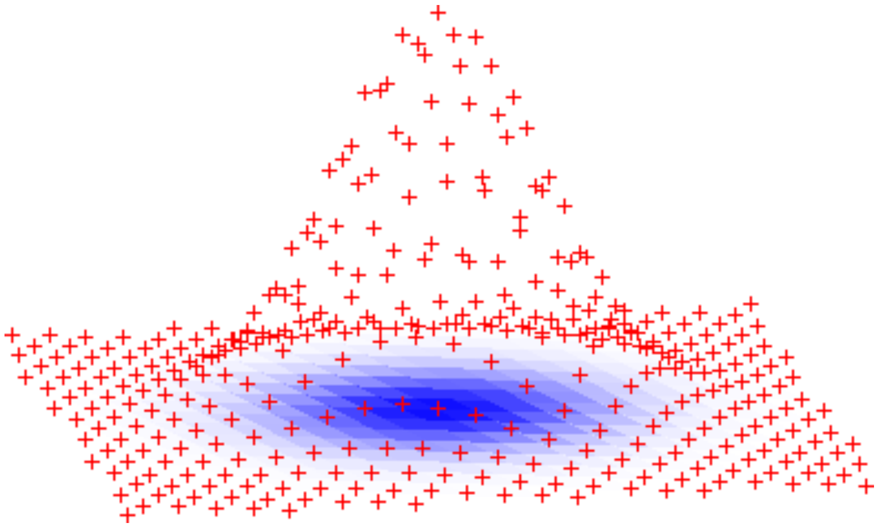
1. To compute a signature, must know the private key
2. To verify a signature, only the public key is needed

Digital Signatures from Lattices

- Generate uniform $vk = \mathbf{A}$ with secret 'trapdoor' $sk = \mathbf{T}$.

Digital Signatures from Lattices

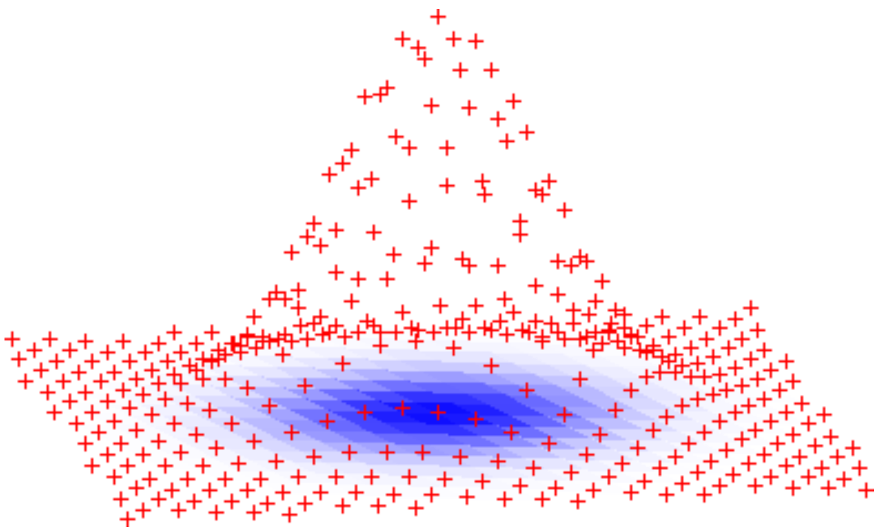
- Generate uniform $vk = \mathbf{A}$ with secret 'trapdoor' $sk = \mathbf{T}$.
- Sign$(\mathbf{T}, \mu)$: use $\mathbf{T}$ to sample a short $\mathbf{z} \in \mathbb{Z}^m$ s.t. $\mathbf{A}\mathbf{z} = H(\mu) \in \mathbb{Z}_q^n$.
  Draw $\mathbf{z}$ from a distribution that reveals nothing about secret key:



Digital Signatures from Lattices

- Generate uniform $vk = \mathbf{A}$ with secret 'trapdoor' $sk = \mathbf{T}$.
- Sign$(\mathbf{T}, \mu)$: use $\mathbf{T}$ to sample a short $\mathbf{z} \in \mathbb{Z}^m$ s.t. $\mathbf{A}\mathbf{z} = H(\mu) \in \mathbb{Z}_q^n$.

  Draw $\mathbf{z}$ from a distribution that reveals nothing about secret key:



- Verify$(\mathbf{A}, \mu, \mathbf{z})$: check that $\mathbf{A}\mathbf{z} = H(\mu)$ and $\mathbf{z}$ is sufficiently short.
- Security: forging a signature for a new message μ^* requires finding short $\mathbf{z}^*$ s.t. $\mathbf{A}\mathbf{z}^* = H(\mu^*)$. This is SIS: hard!


We saw some foundations…
Also promised opportunities…

Lots and lots of questions

- Multilinear (even bilinear) maps from lattices?
- Non-Interactive Key Exchange?
- Efficient Threshold Signatures?
- Witness Encryption?

Bilinear Maps

Let G_1, G_2, G_T be groups of prime order and g_i denote the generator of G_i

$$e\colon\ \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$$

$$e\left(g_1^a, g_2^b\right) = e(g_1, g_2)^{ab}$$

Hardness Assumption (roughly): Adversary can only compute pairings, take linear combinations and test if output is zero

Lattice version?

Non-Interactive Key Exchange

pk_1, pk_2, pk_3, pk_4

$(pk_1, sk_1) \leftarrow KeyGen$

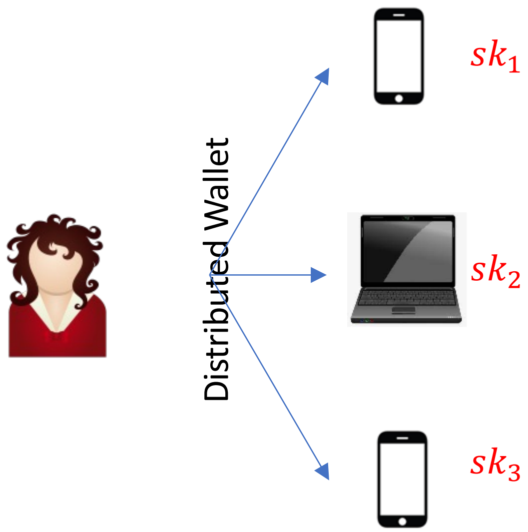
$(pk_2, sk_2) \leftarrow KeyGen$

$(pk_3, sk_3) \leftarrow KeyGen$

$(pk_4, sk_4) \leftarrow KeyGen$

- Derive a shared key K_{1234}

Post Quantum Threshold Signatures

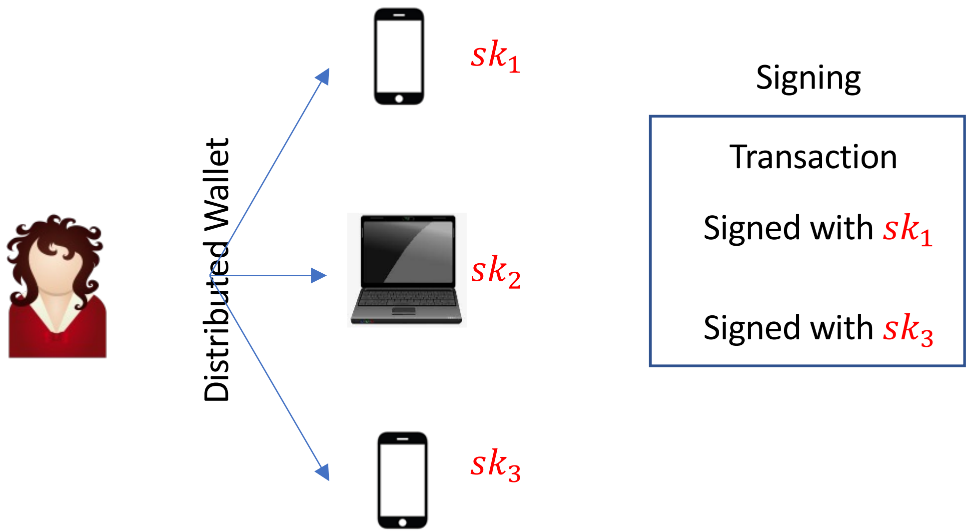


Signing

Verification

Common verification key vk

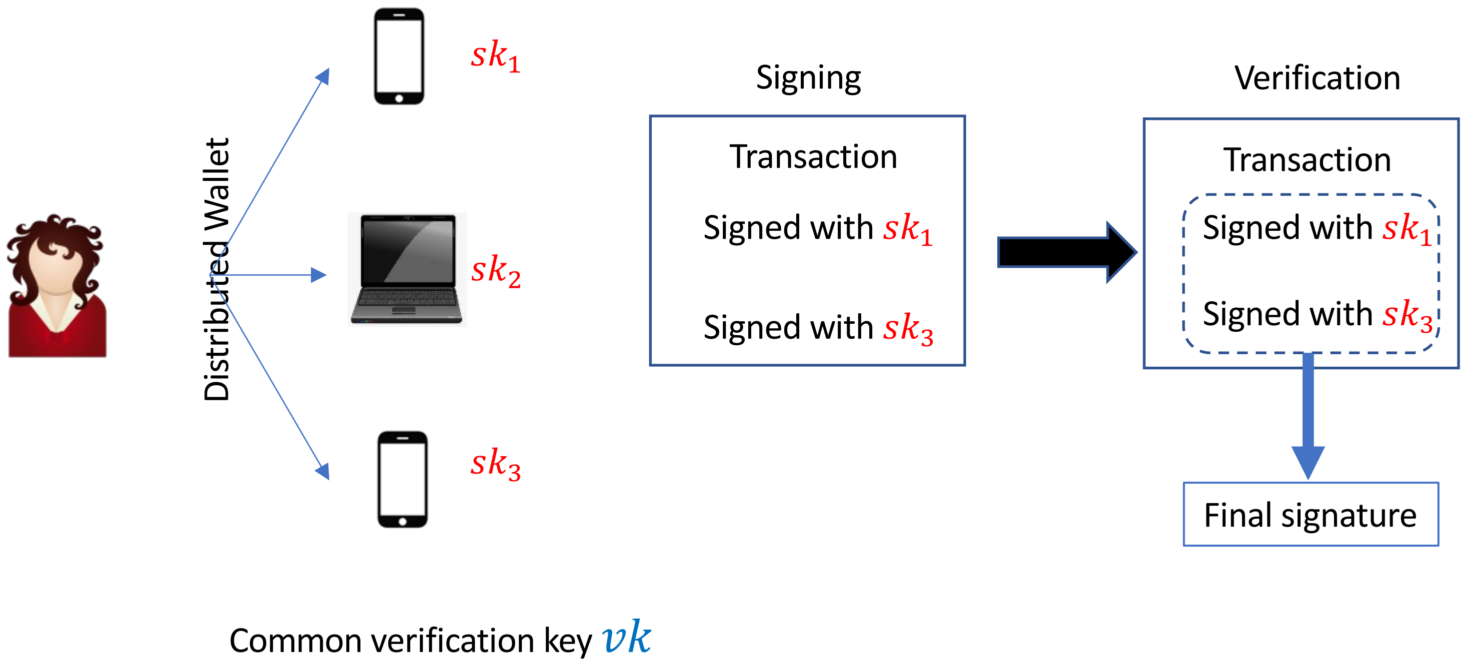
Post Quantum Threshold Signatures



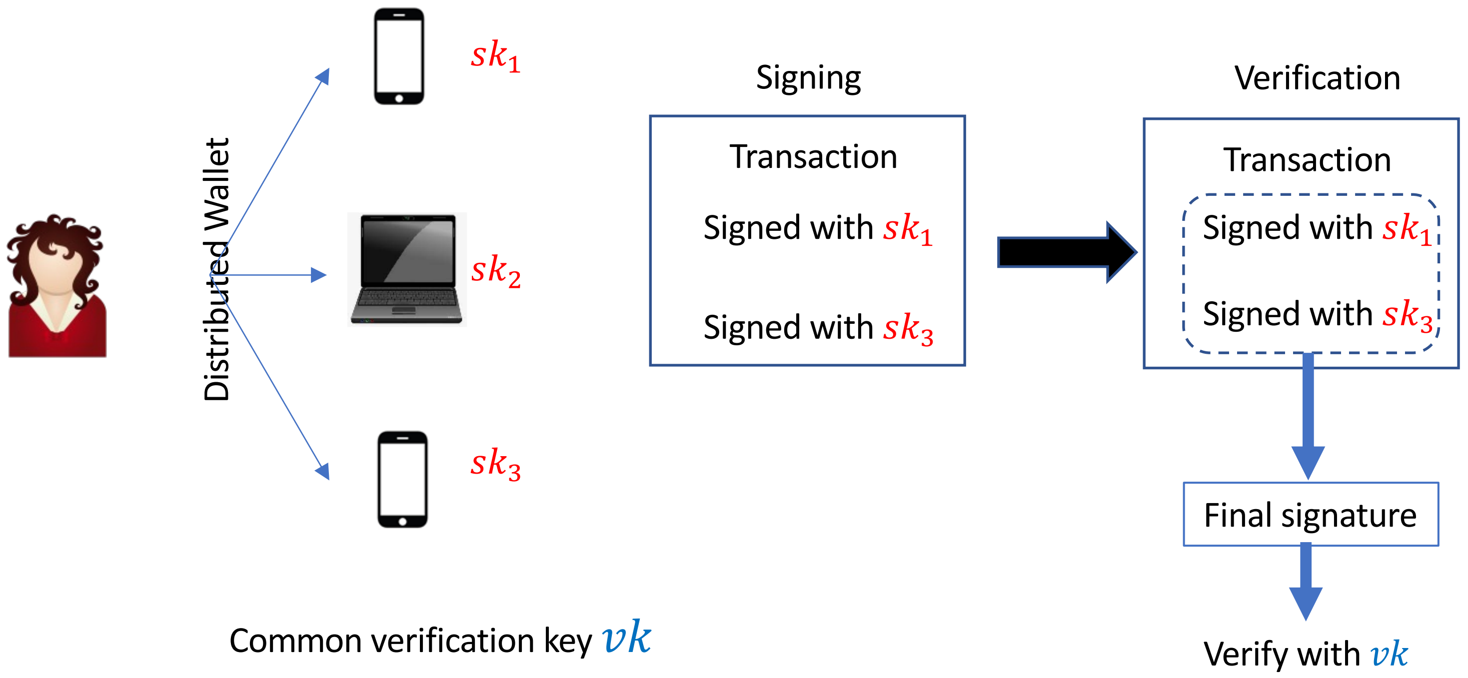
Verification

Common verification key vk

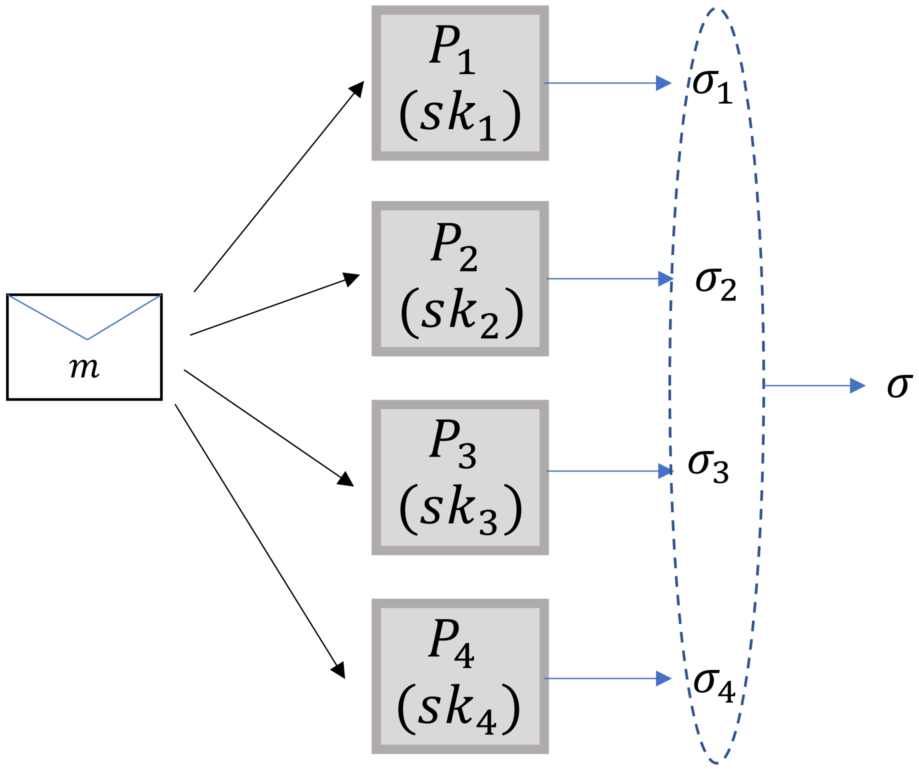
Post Quantum Threshold Signatures



Post Quantum Threshold Signatures



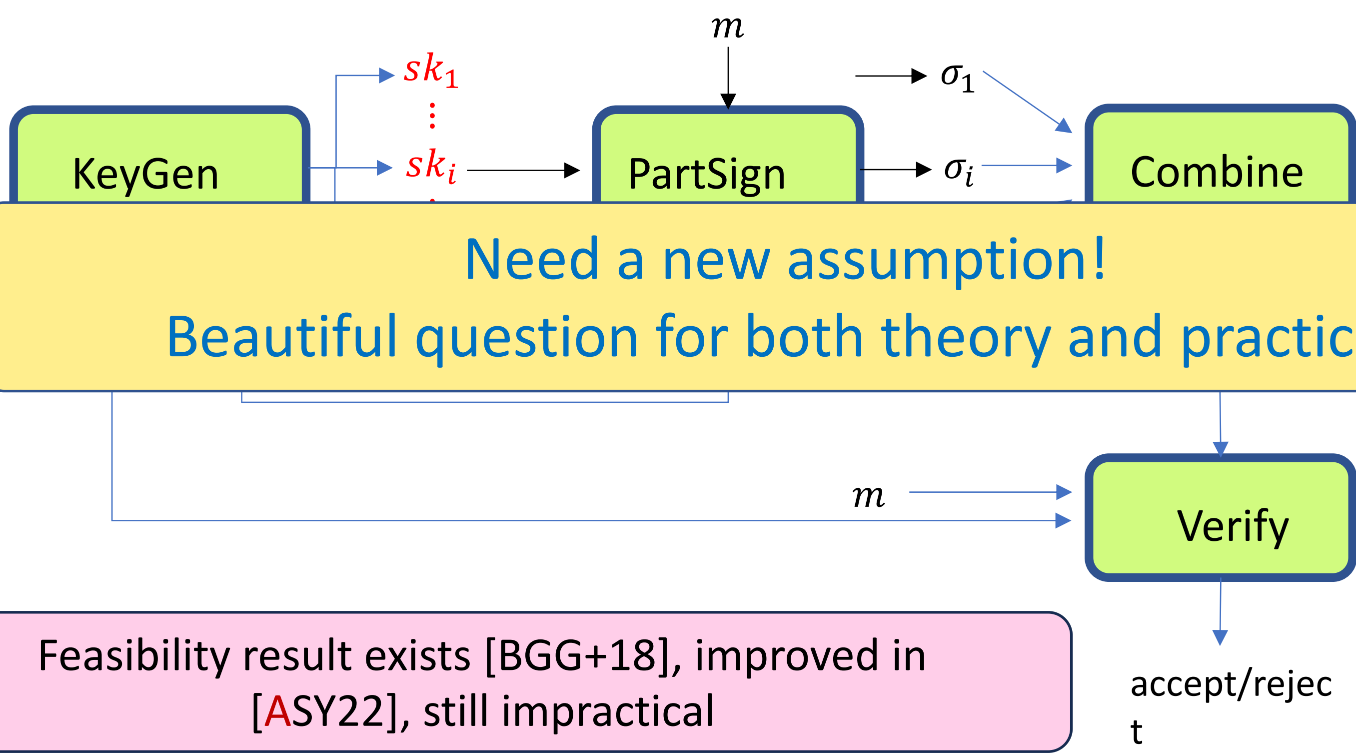
Post Quantum Threshold Signatures



Correctness – Signature generated from contribution of a valid set of participants should verify.

Security - Any invalid set of participants should not be able to generate a valid signature

Post Quantum Threshold Signatures



Need a new assumption!

Beautiful question for both theory and practice!

Feasibility result exists [BGG+18], improved in [ASY22], still impractical

Witness Encryption

Encrypt against NP statement, Decrypt with witness!

- Encrypt (x, m) → ct
- Decrypt (ct, w) → m iff w is witness for statement x

Currently no construction from good assumption!

Summary

- Post Quantum Crypto: Intro
- Basics of Lattices
- Hard Problems on Lattices
- Public Key Encryption
- Digital Signatures
- Taste of open questions

Thank You

Images Credit: Hans Hoffman

Slides Credit: Daniele Micciancio, Chris Peikert