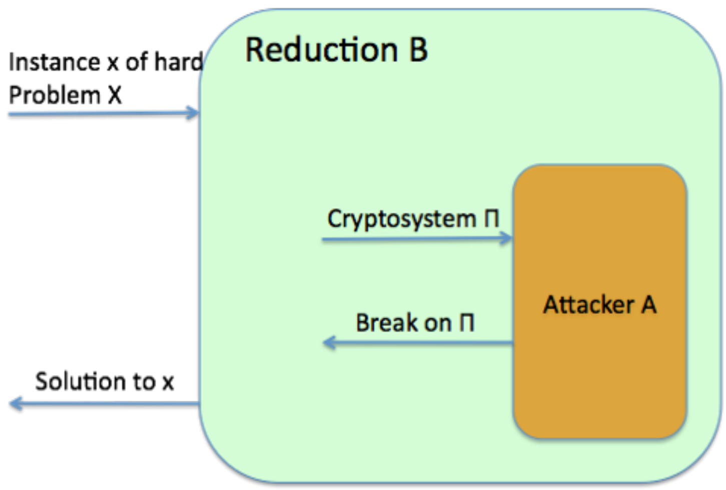
Post Quantum Cryptography: An Overview

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?

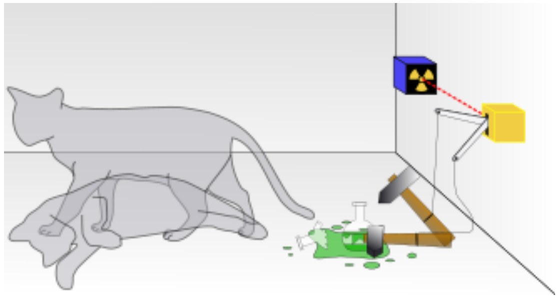
The Cryptographic Adversary

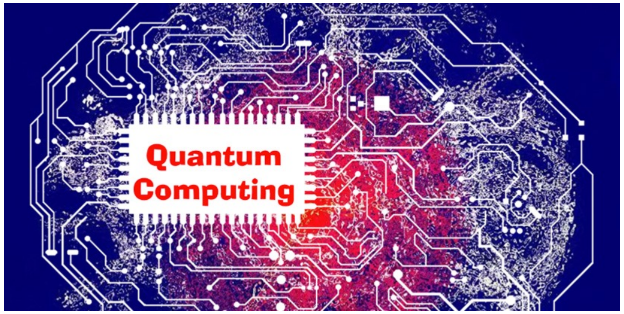
- Adversary in cryptography normally modeled by a classical computer.
- Typical guarantee is that unless the adversary can solve hard problem, attack takes more than age of universe (in CPU cycles)
- Robust to type of computer (mobile/laptop/supercomputer)
- What if the attacker is quantum?

Quantum Computers

Fundamentally New Paradigm of Computing!

- Computers that use laws of quantum rather than classical physics
- Allow transformation of memory to quantum superposition of all possible classical states
- May allow exponential speedups
- Most current cryptography relies on hardness of factoring, discrete log: **broken** if quantum computers are realized





Is this threat real?

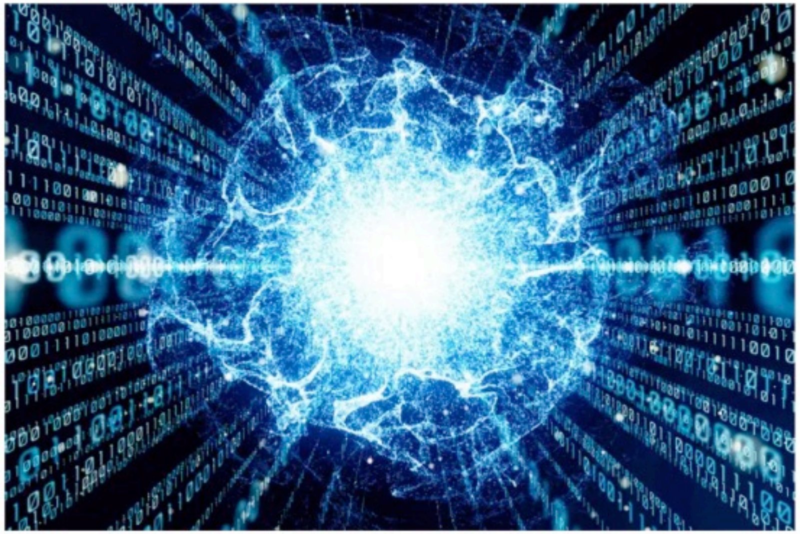
In short: YES!

- National Institute of Standards and Technology (NIST), initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key crypto algorithms
- Significant global research effort

Google claims it has finally reached quantum supremacy

PHYSICS 23 September 2019

By Chelsea Whyte

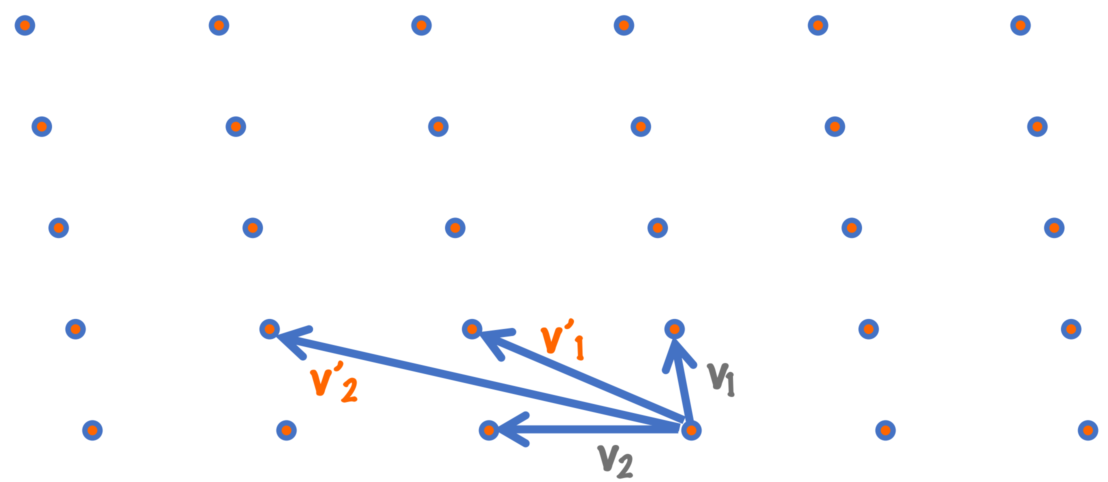


Google's demonstration reportedly involved checking a series of binary numbers were truly random
iStock / Getty Images Plus

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

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

Exciting New Applications

Encrypted Computation
Personalised Medicine

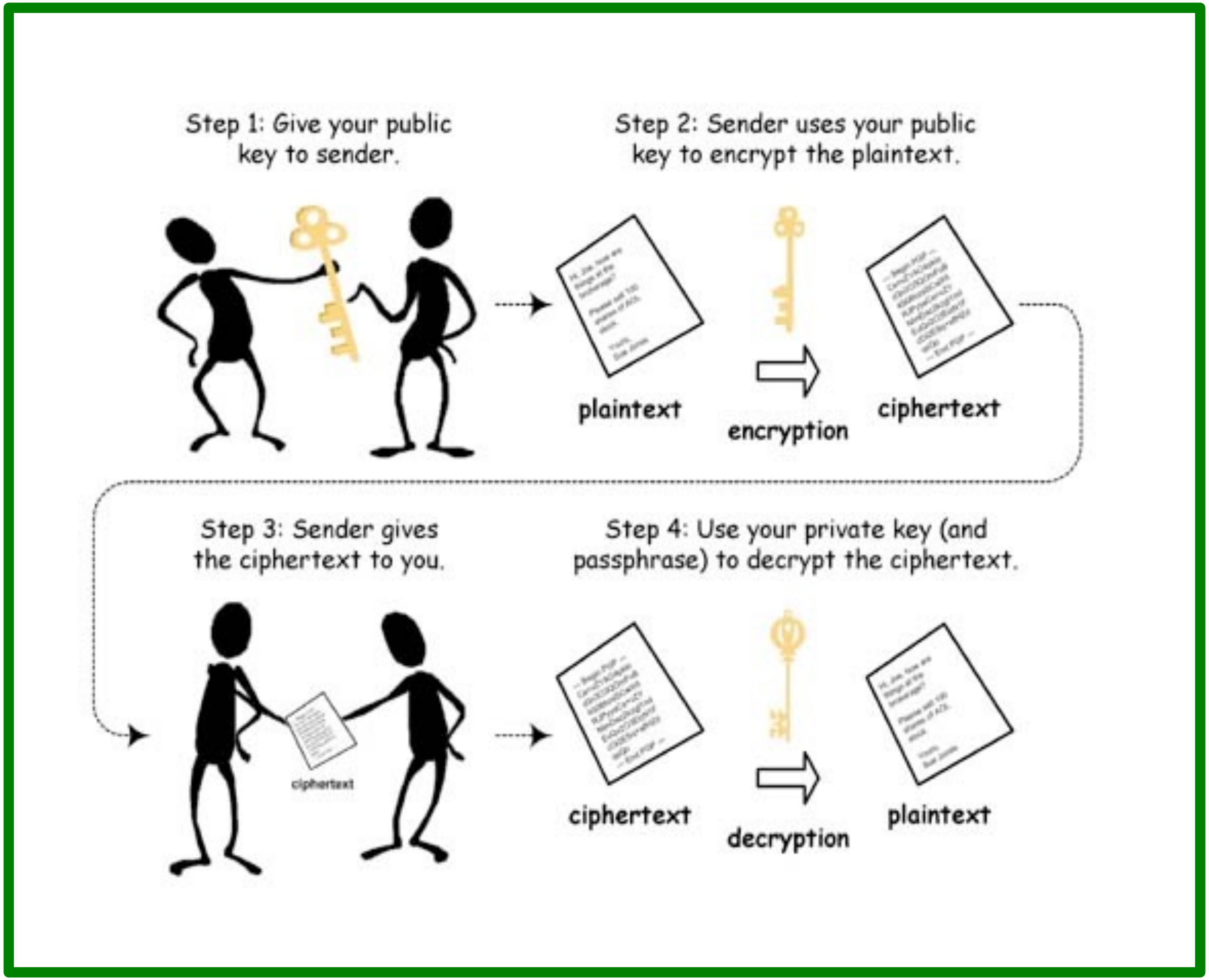
"The dream for tomorrow's medicine is to understand the links between DNA and disease — and to tailor therapies accordingly. But scientists have a problem: how to keep genetic data and medical records secure while still enabling the massive, cloud-based analyses needed to make meaningful associations."

"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



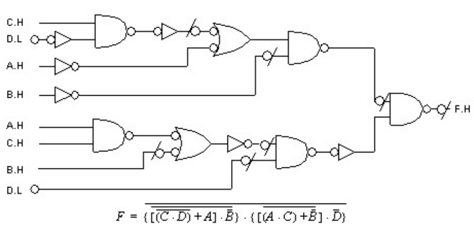
PKE does not suffice!

- 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

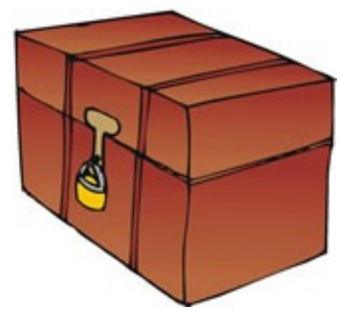
Functional Encryption!

Secret Keys
for functions *F*



+

Ciphertexts
for inputs *x*



Decryption recovers F(x)

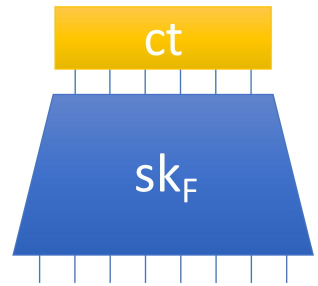
F : Age distribution of people with lung cancer
X : particular user's disease profile

Encryption with Partial Decryption Keys

Encrypt (x):

Keygen(F):

Decrypt (sk_F, ct):



y = F(x)

Security:

Adversary possessing keys for multiple circuits F_i cannot distinguish $\mathrm{Enc}(x_0)$ from $\mathrm{Enc}(x_1)$ unless $F_i(x_0) \neq F_i(x_1)$

Functional Encryption [SW05,BSW11]

Personalized Medicine?

Encrypt
input = genomic data of users

ct(Anil)

ct(Prabha)

ct(Vaibhav)

ct(Shweta)

Keygen
input: some medical research algo

Decrypt (sk$_F$, ct):

ct(Anil) ct(Prabha)

ct(Vaibhav) ct(Shweta)

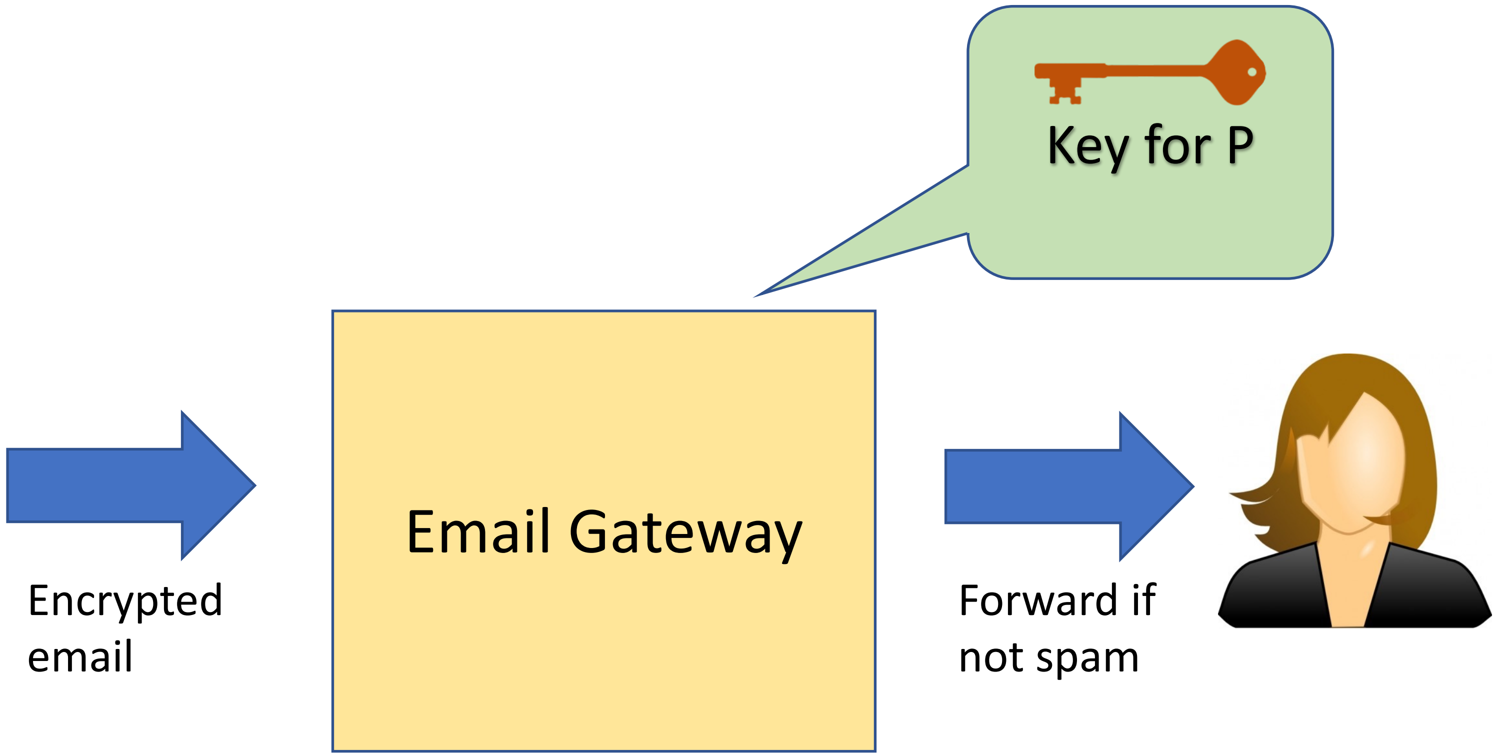
y = F(x)

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

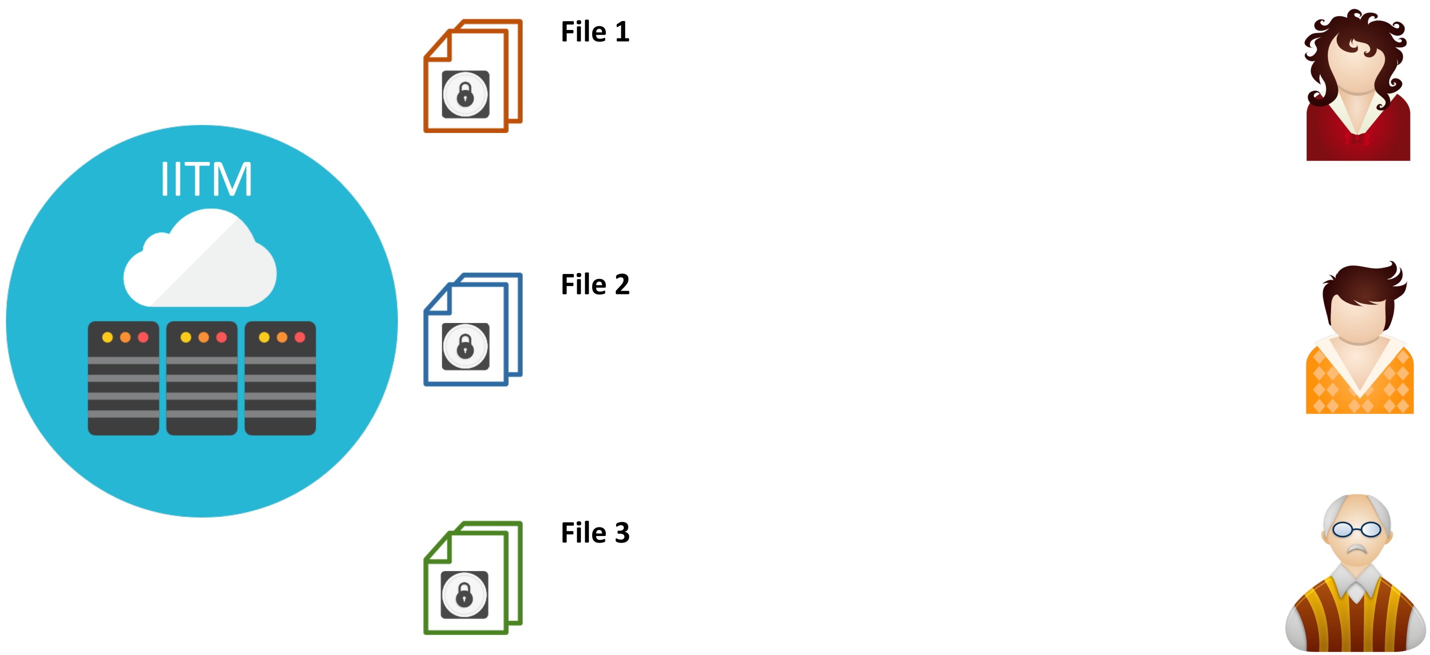
Functional Encryption [SW05,BSW11]

Spam Detection on Encrypted Email

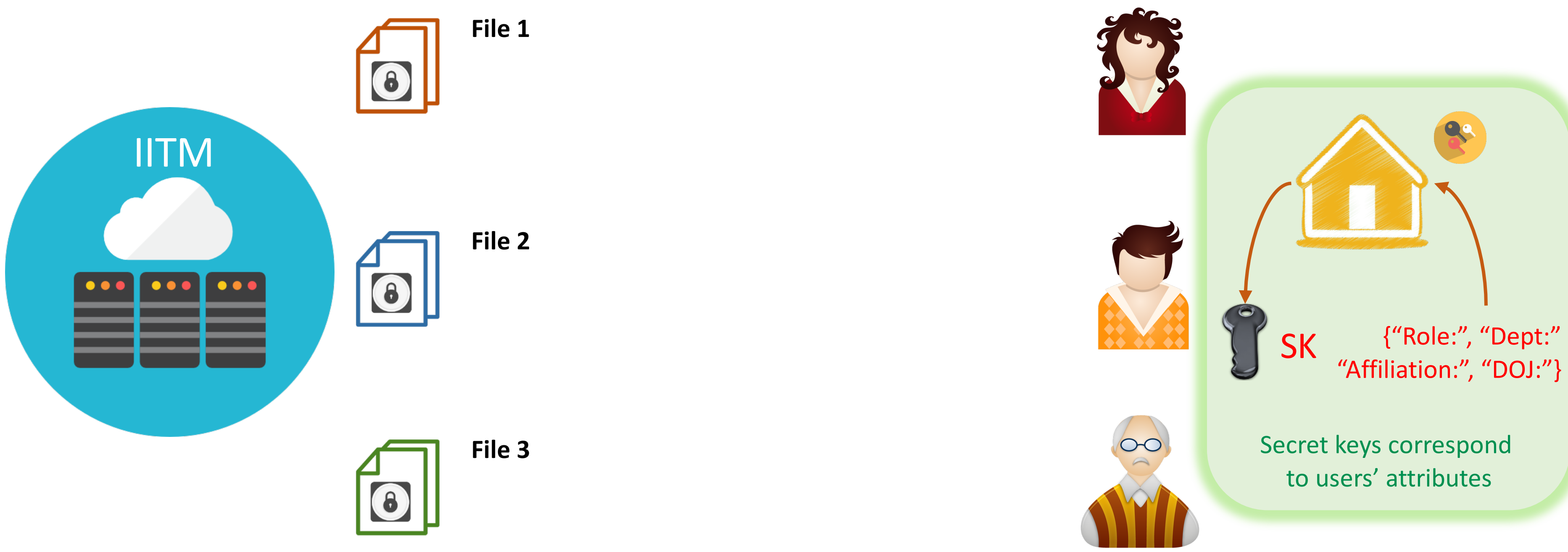
Say we have a program P to detect spam on unencrypted email.



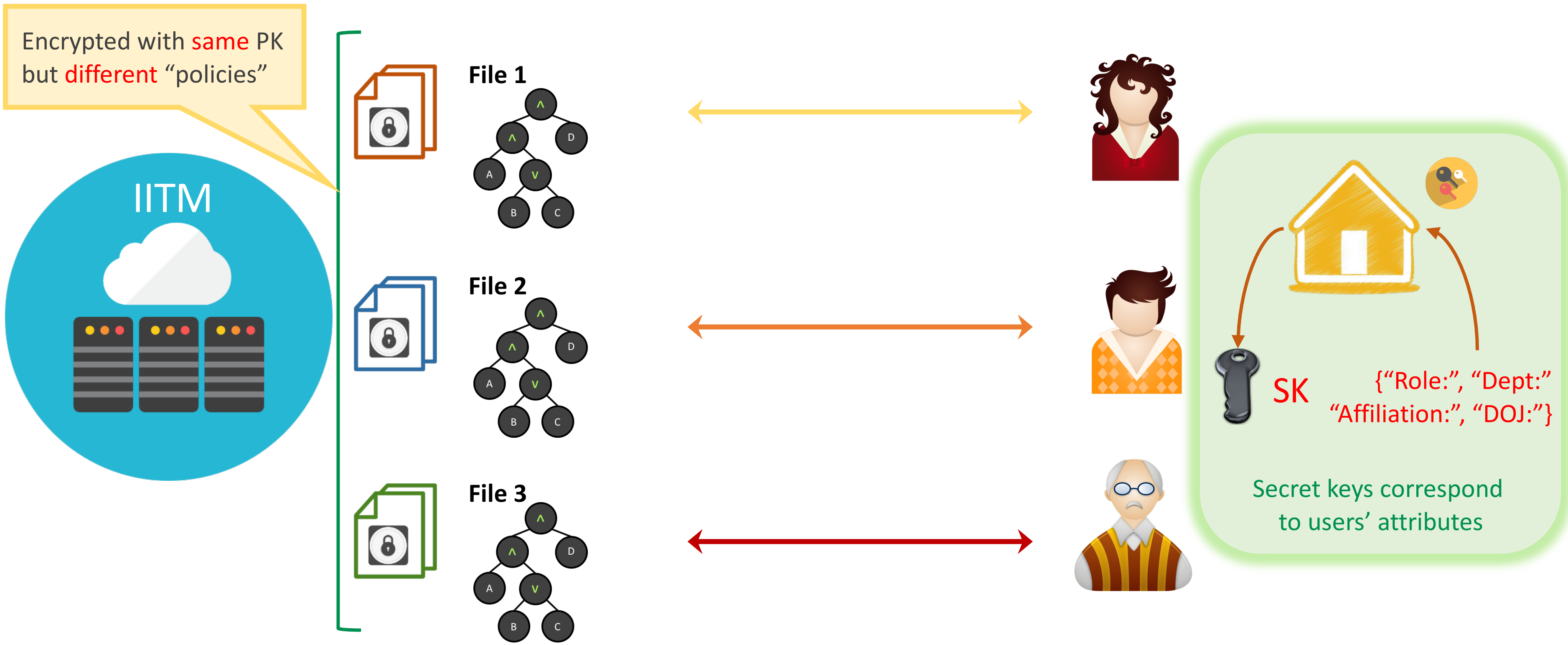
Attribute based Encryption
[SW05, GPSW06]



Attribute based Encryption
[SW05, GPSW06]

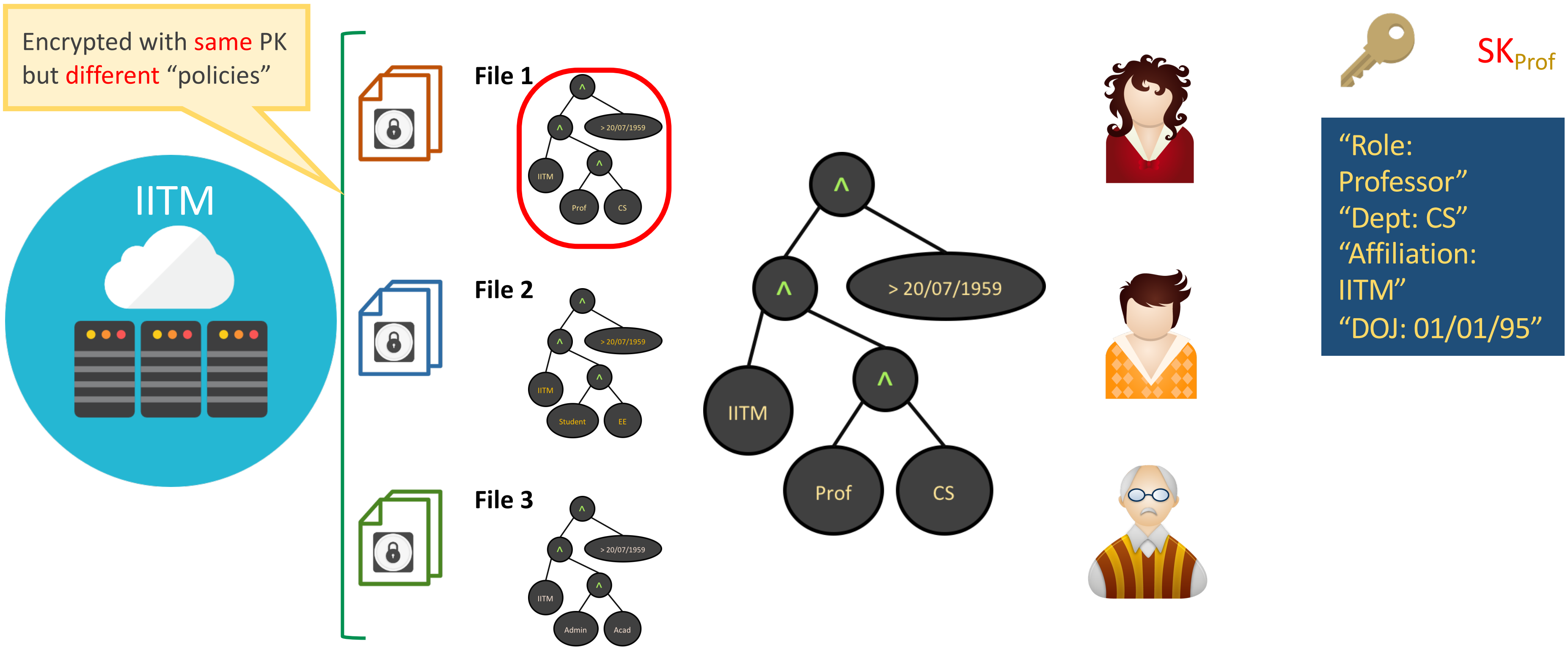


Attribute based Encryption
[SW05, GPSW06]



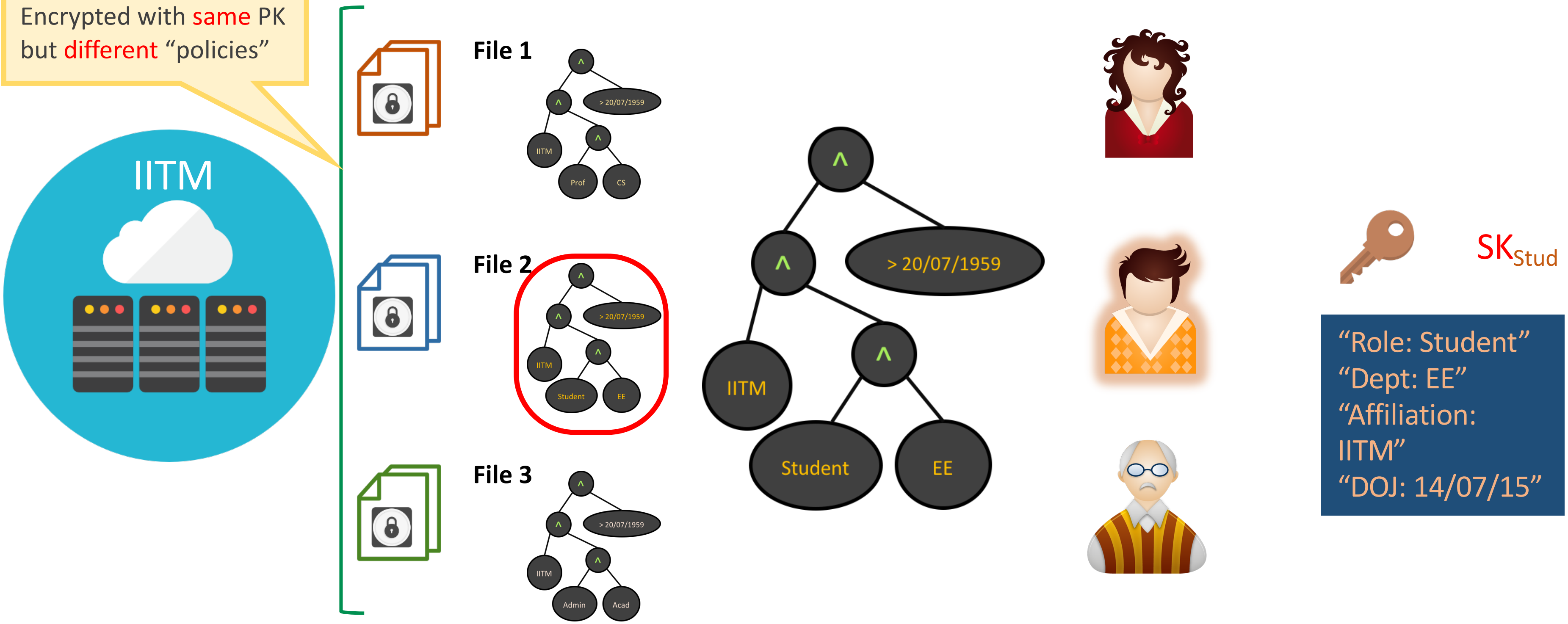
Attribute based Encryption

[SW05, GPSW06]



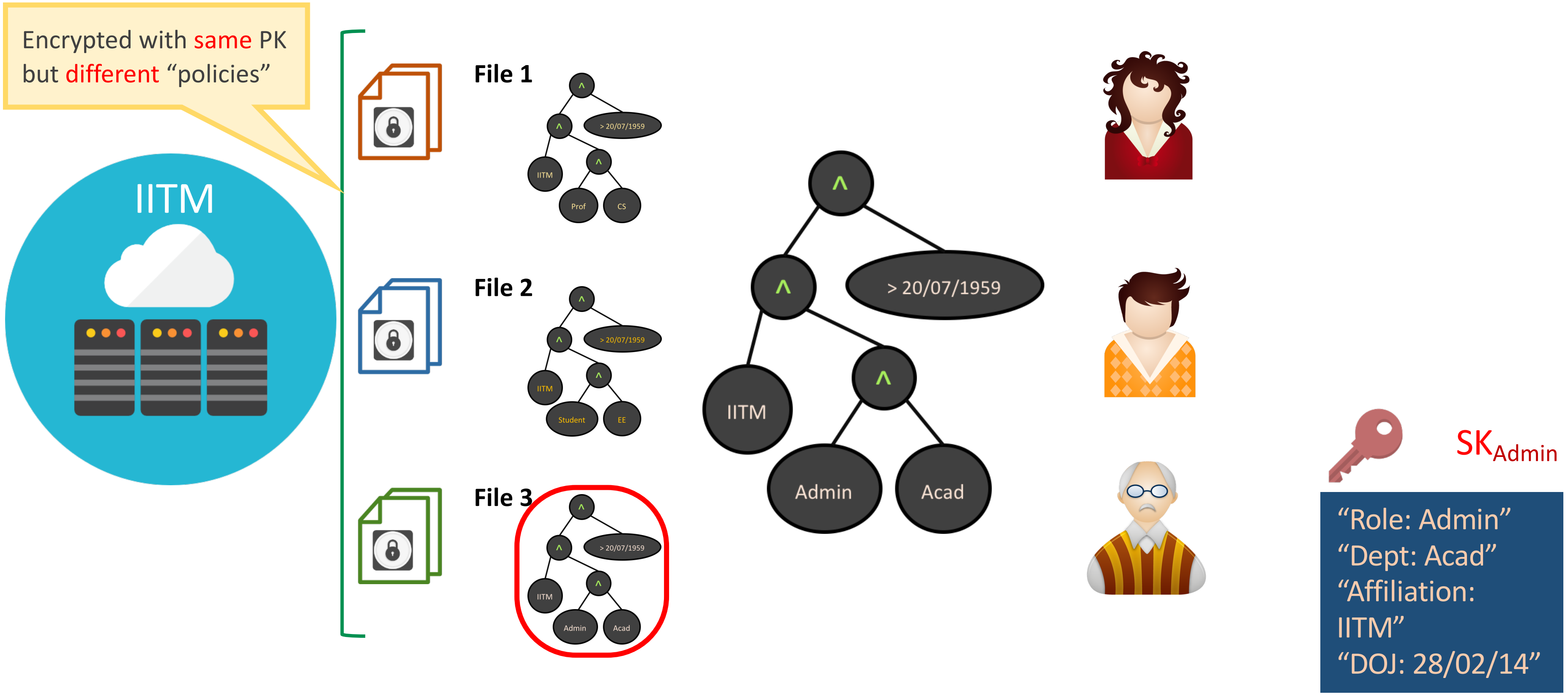
Attribute based Encryption
[SW05, GPSW06]

Encrypted with same PK but different "policies"

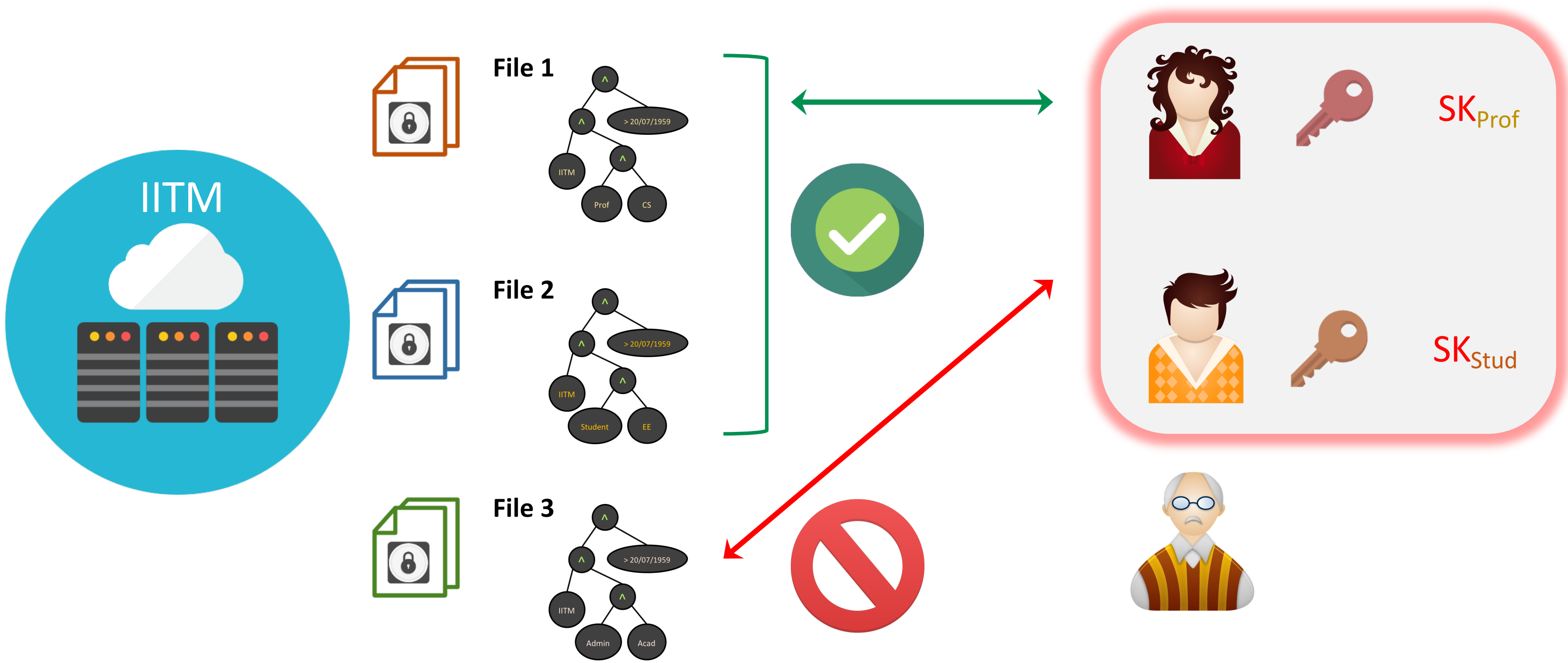


Attribute based Encryption

[SW05, GPSW06]

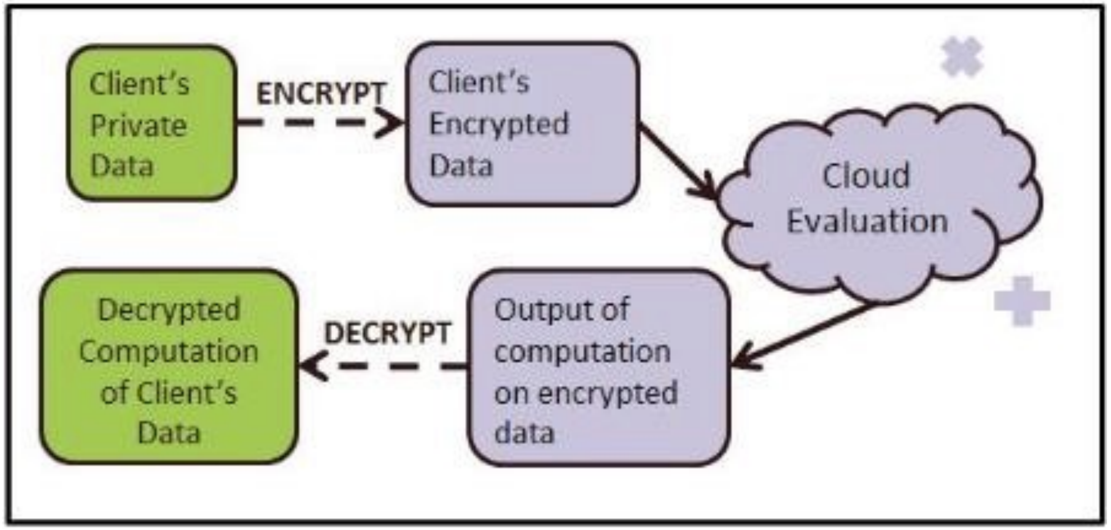


Attribute based Encryption
[SW05, GPSW06]



Fully Homomorphic Encryption

[G09, BV11, BGV12, GSW13…]



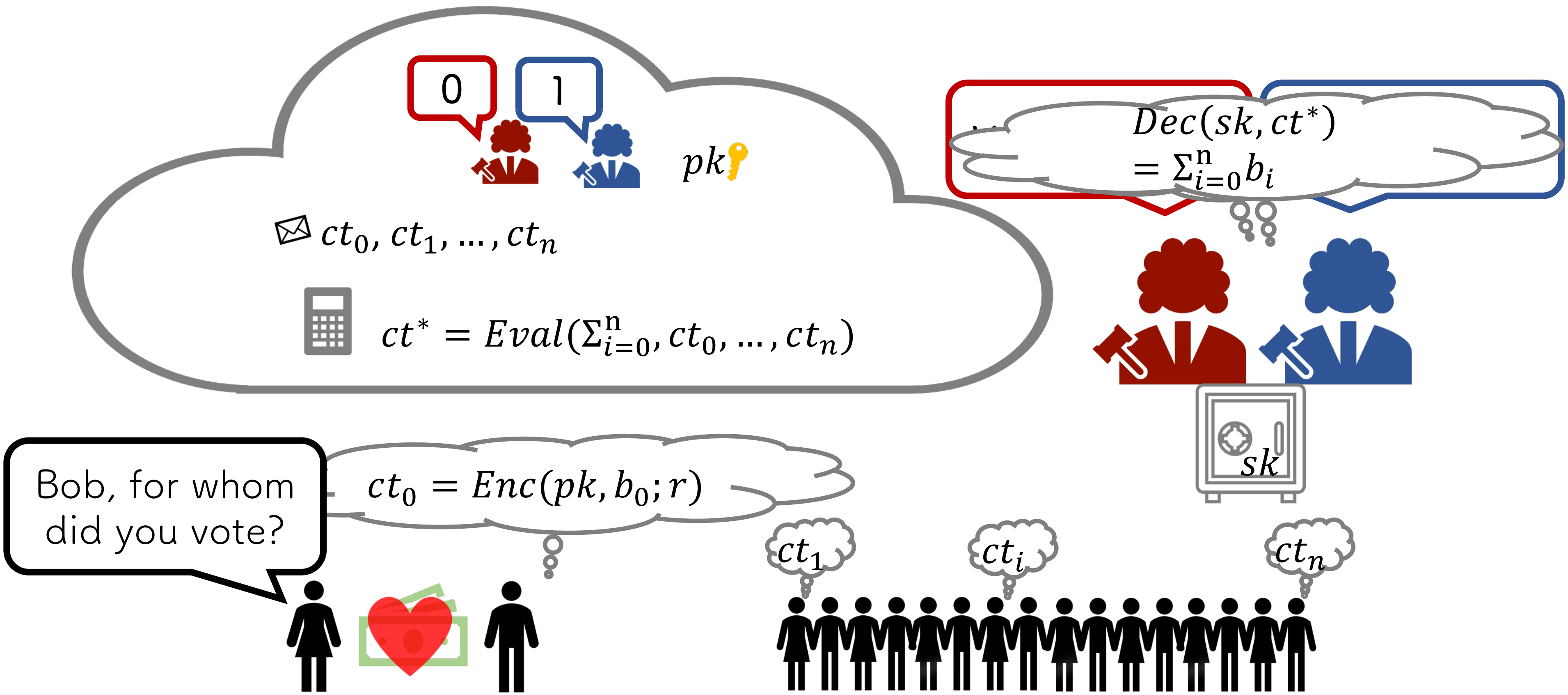
Expressive Functionality: Supports arbitrary circuits

Compact ciphertext, independent of circuit size

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

* : roughly

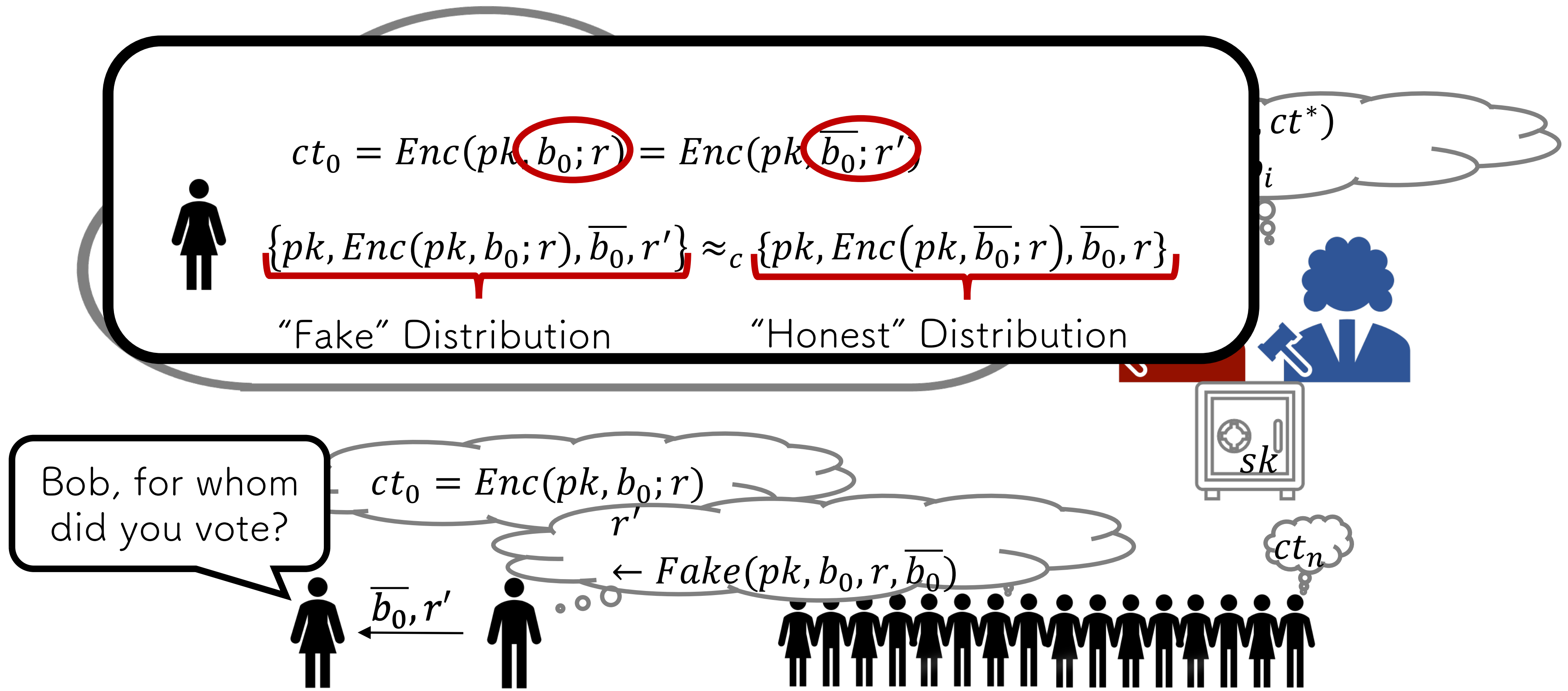
Deniable FHE [AGM21]

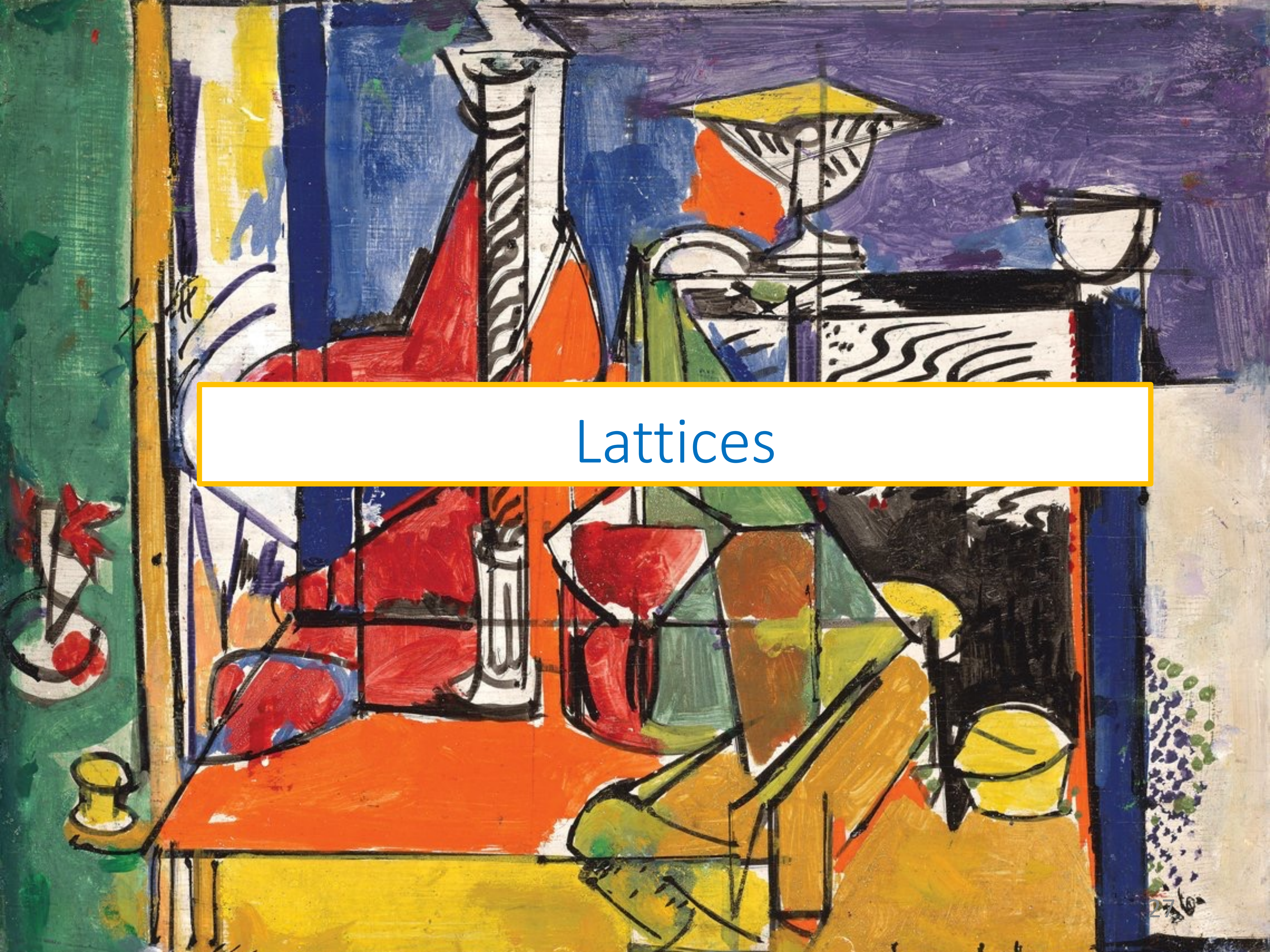


Deniable FHE [AGM21]

$$ct_0 = Enc(pk, b_0; r) = Enc(pk, \overline{b_0}; r')$$

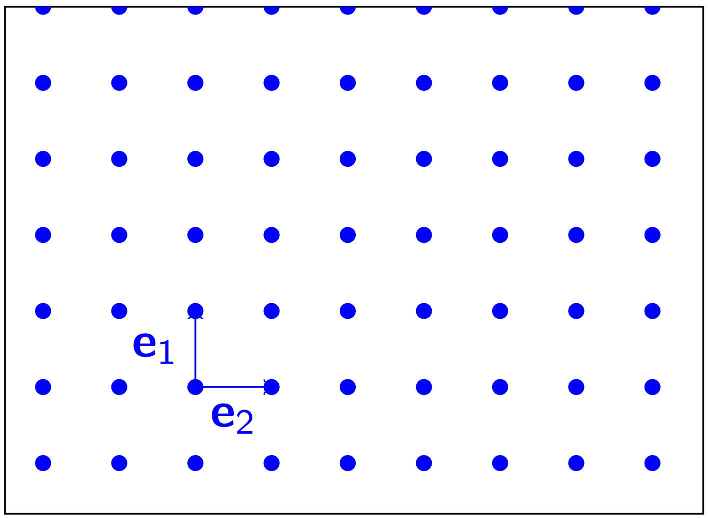
$$\underbrace{\{pk, Enc(pk, b_0; r), \overline{b_0}, r'\}}_{\text{“Fake” Distribution}} \approx_c \underbrace{\{pk, Enc(pk, \overline{b_0}; r), \overline{b_0}, r\}}_{\text{“Honest” Distribution}}$$





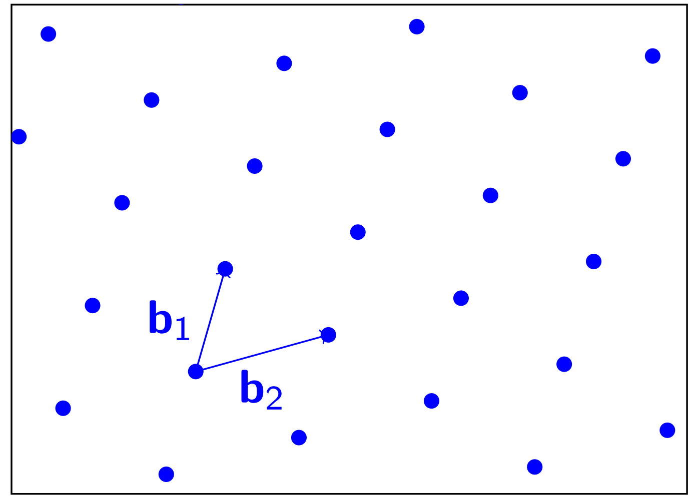
Lattices

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

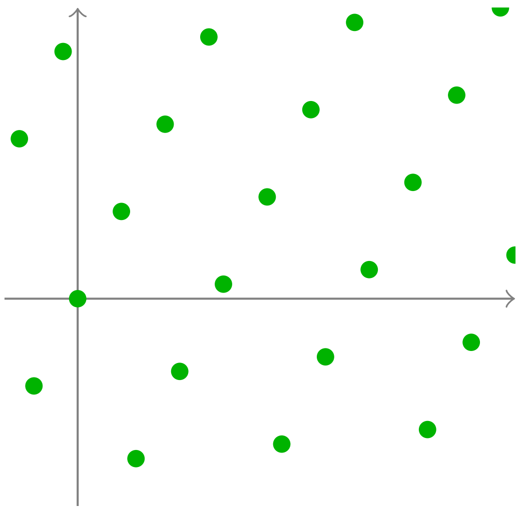
Lattices and Bases

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

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

The same lattice has many bases

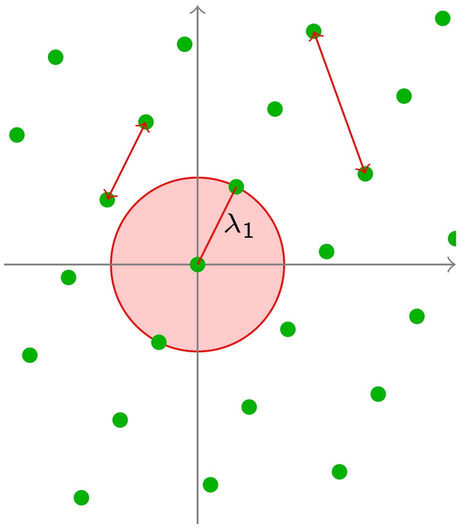
$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{c}_i \cdot \mathbb{Z}$$



Minimum Distance and Successive Minima

- Minimum distance

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



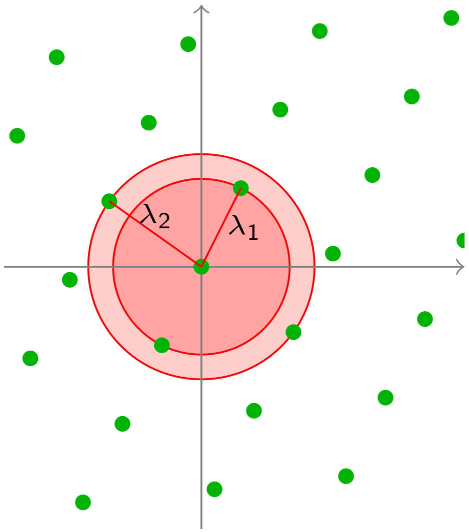
- Successive minima ($i = 1, \ldots, n$)

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

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 $(i = 1, \ldots, n)$

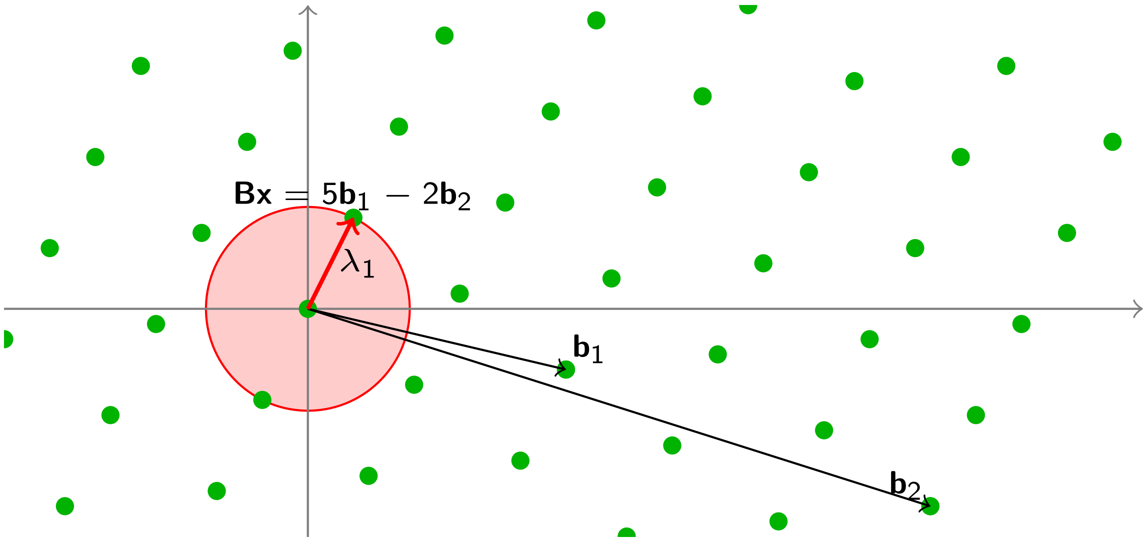
$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$

- 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)

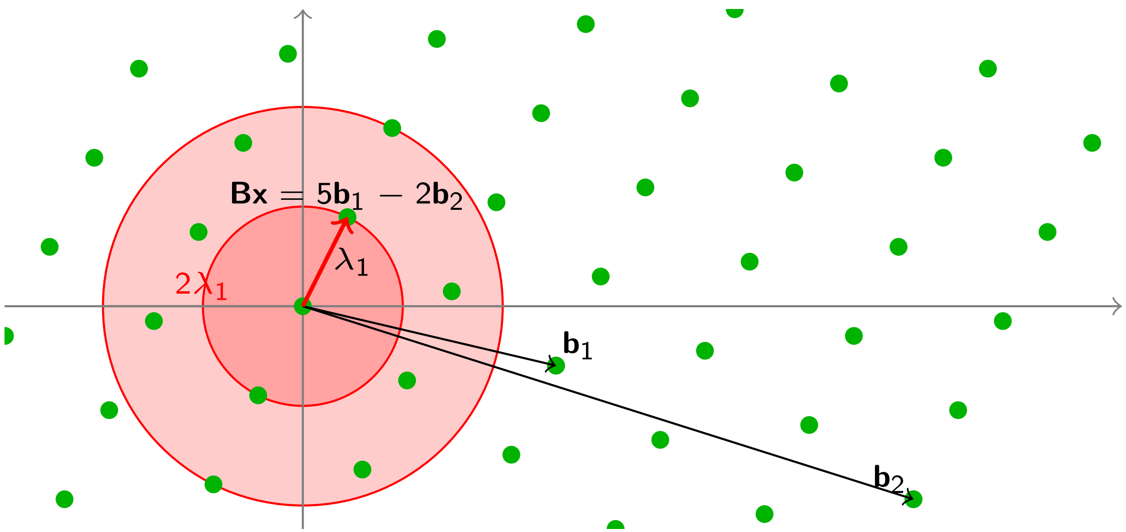
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Approximate Shortest Vector Problem

Definition (Shortest Vector Problem, SVP_γ)

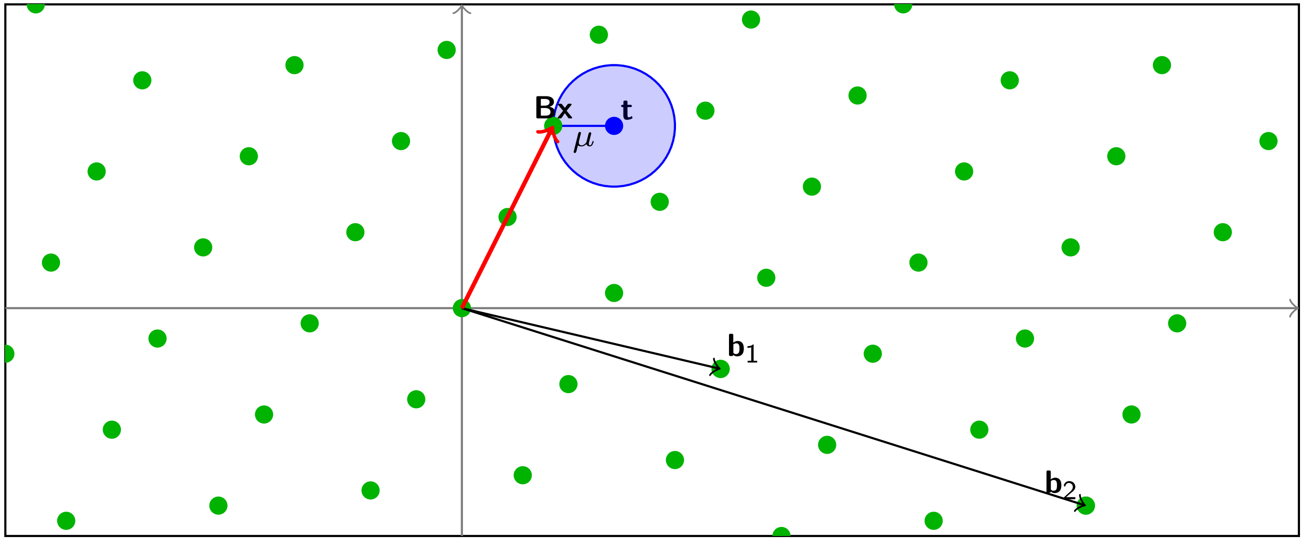
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \gamma\lambda_1$



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

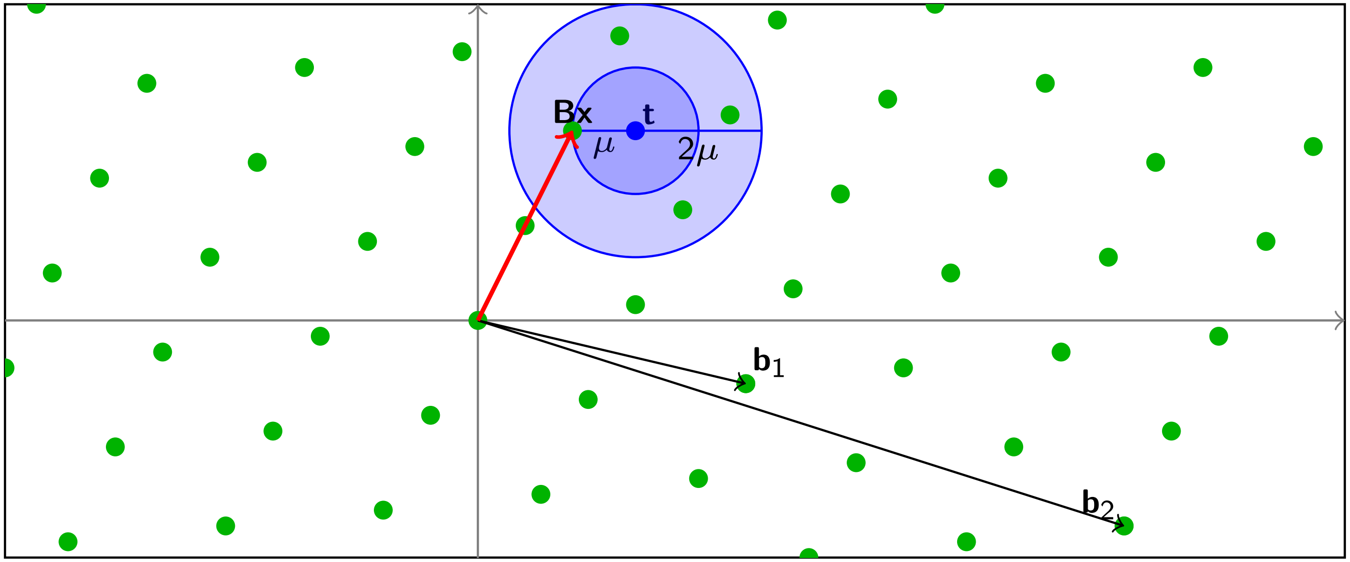
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \mu$ from the target



Approximate Closest Vector Problem

Definition (Closest Vector Problem, CVP$_\gamma$)

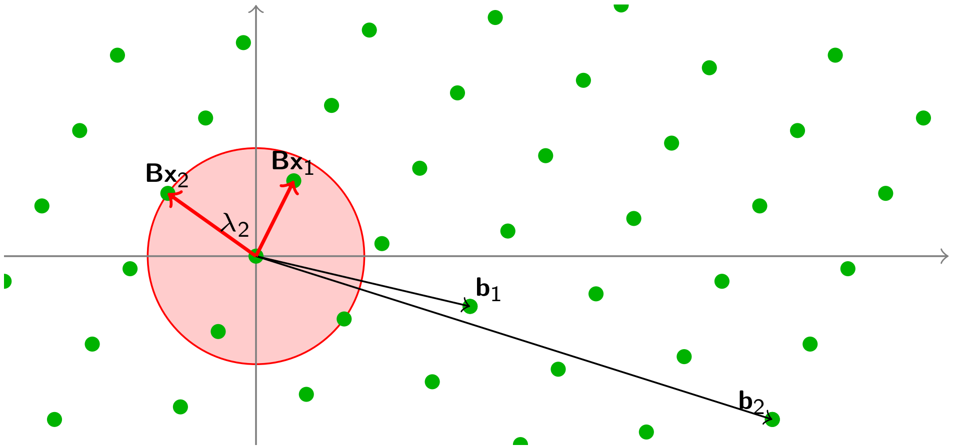
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx}-\mathbf{t}\| \le \gamma\mu$ from the target



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

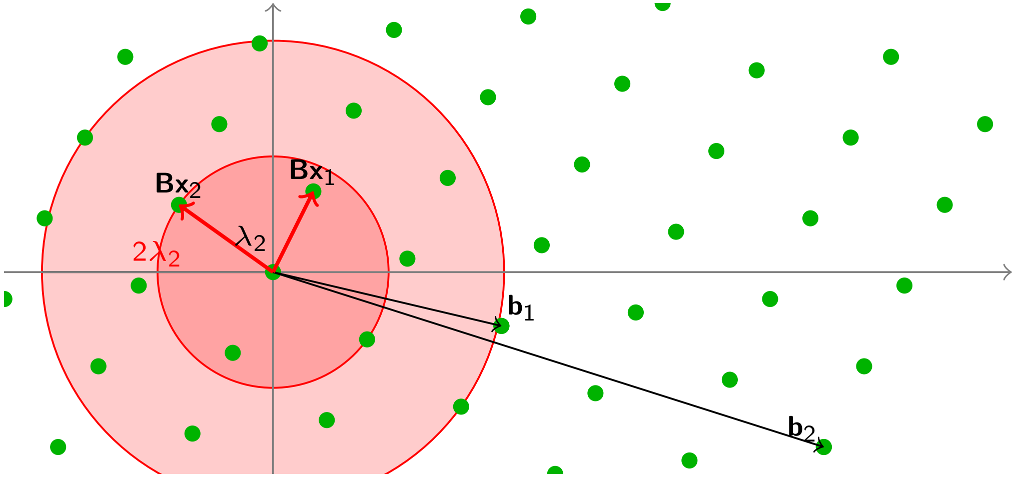
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



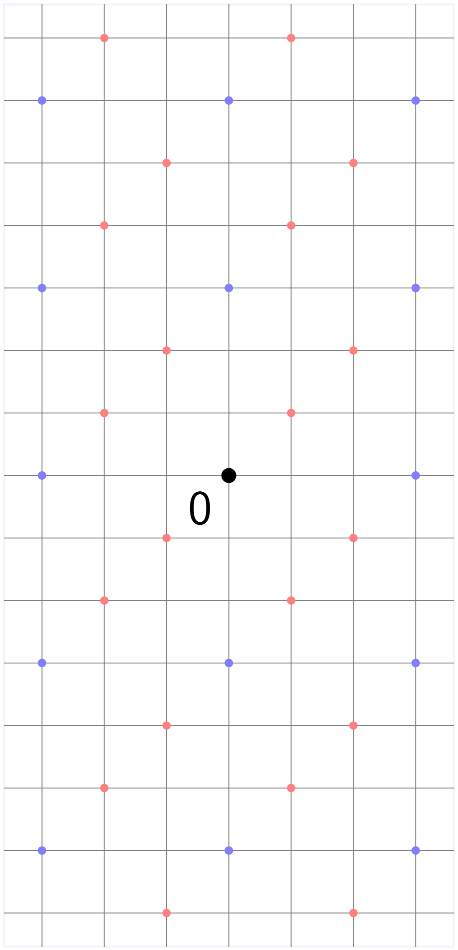
Approximate Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP_γ)

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



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.

Definition (q-ary lattice)

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

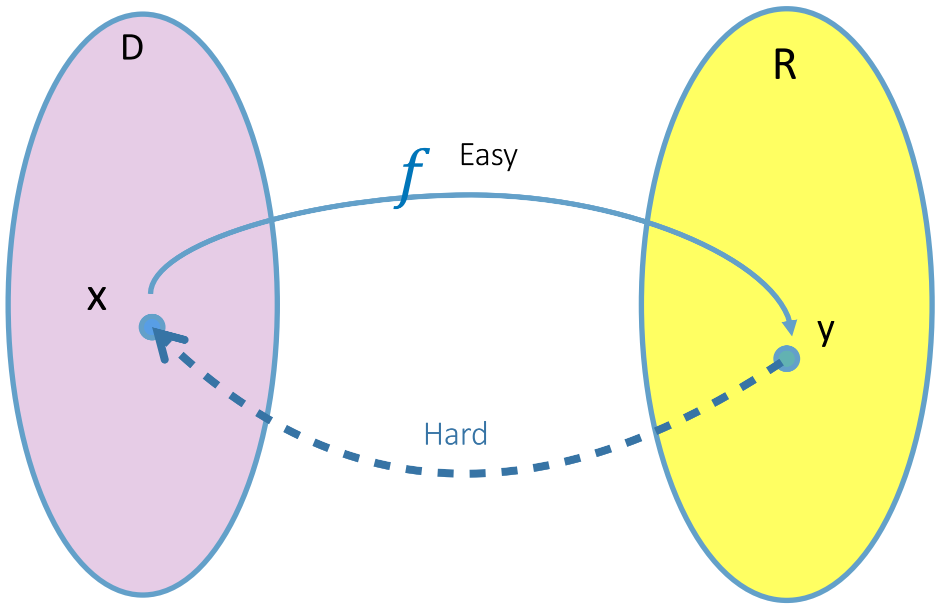
Examples (for any $\mathbf{A} \in \mathbb{Z}_q^{n \times d}$)

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

Building Cryptography

One Way Functions

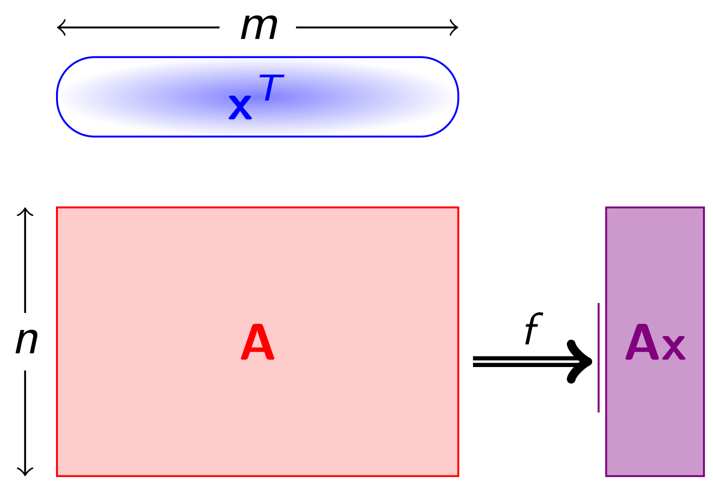
$f\colon D \to R$, **One Way**



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: $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$

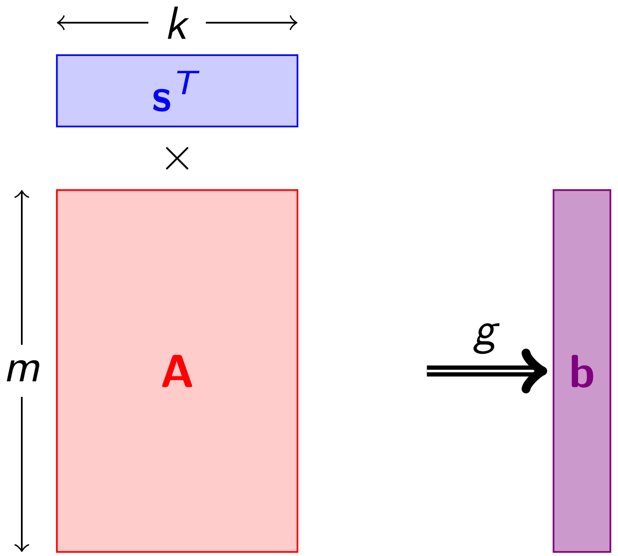


Theorem (A'96)

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.

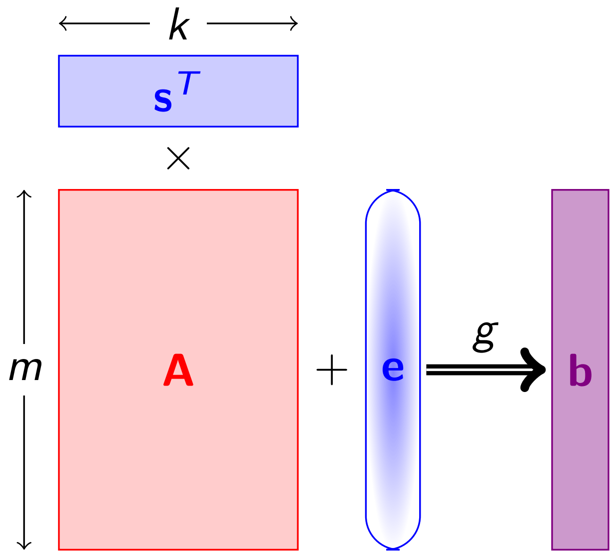
Regev's One Way Function

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



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)

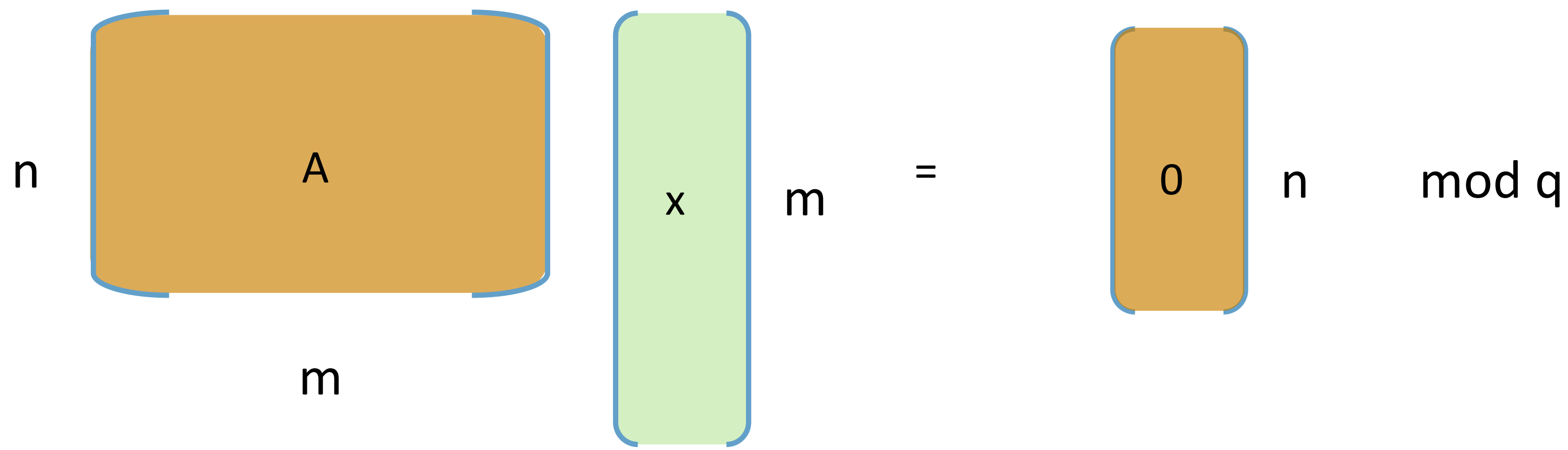
The function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ is hard to invert on the average, assuming SIVP is hard to approximate in the worst-case.

Short Integer Solution Problem

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

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 , $b_2 = \langle a_2, s \rangle + e_2$

$\vdots$

a_m , $b_m = \langle a_m, s \rangle + e_m$

vs

a'_1 , b'_1

a'_2 , b'_2

$\vdots$

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 $\in Z_q$

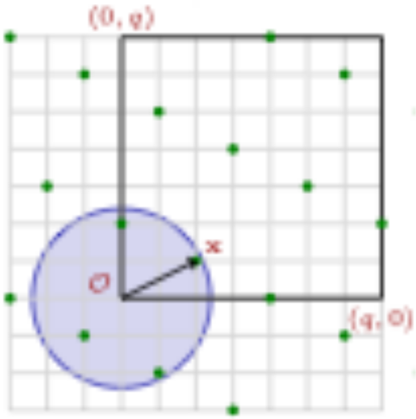
Recap:Lattice Based One Way Functions

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

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…]

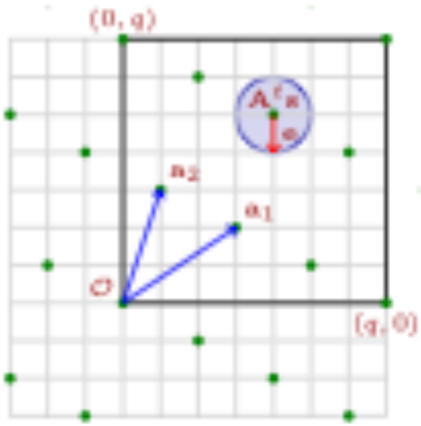


Image Credit: MP12 slides

Public Key Encryption [Regev05]

- Recall $A\,(e) = u \bmod q$ 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
- Secret: e, Public : A, u

$$A \; e \equiv u \mod q$$

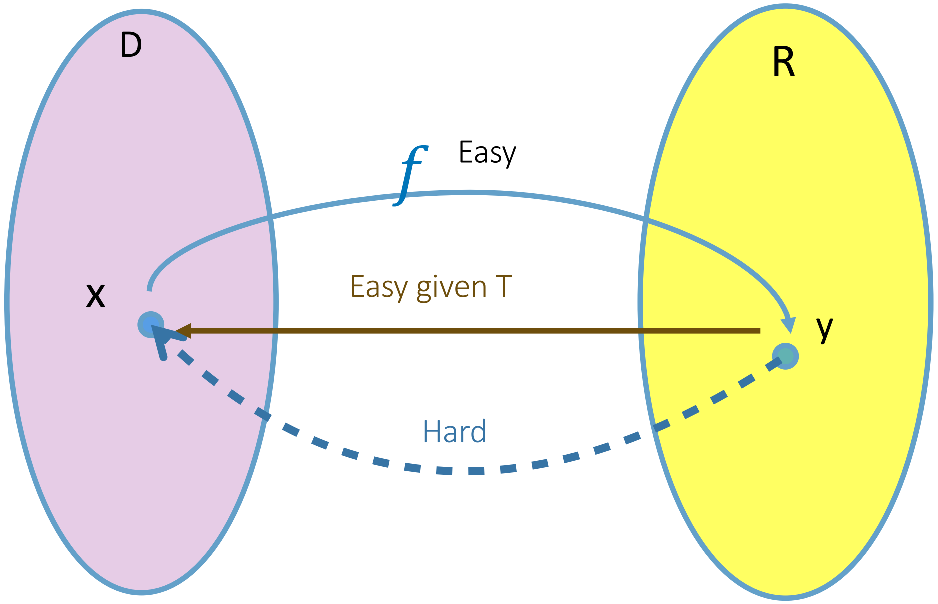
- By SIS problem, hard to find short e
- By LWE problem, ciphertext appears random
  - $c_0 = A^T s$ + noise, looks like random
  - $c_1 = u^T s$ + noise + msg , looks like random + msg
  - Hence hides message “msg”

For Signatures, need Lattice Trapdoors

Trapdoor Functions

Generate (f, T)

$f: D \rightarrow R$, One Way



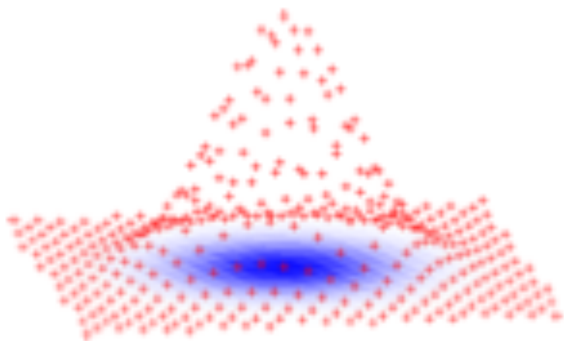
We will construct trapdoor functions from two lattice problems

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



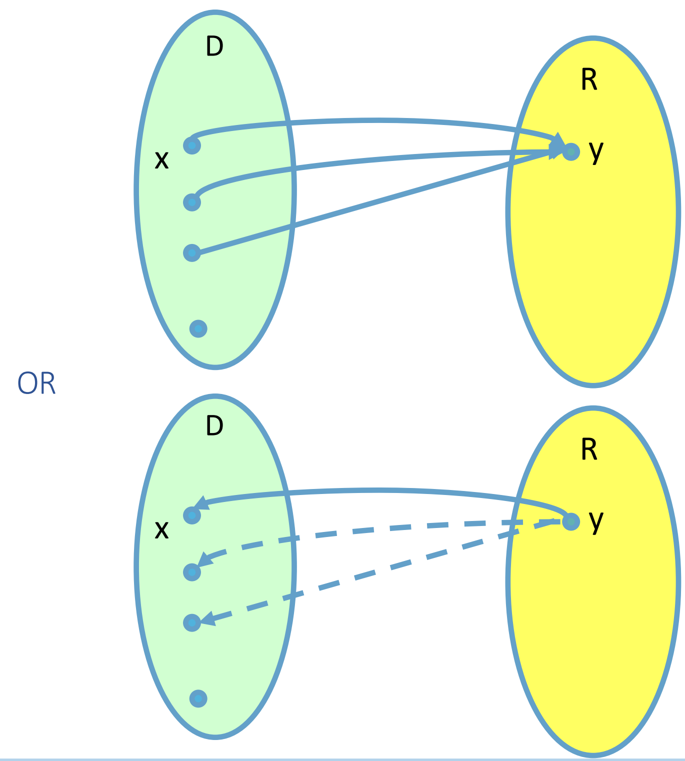
- (
- F

$nodq$

Latter distribution needs lattice trapdoors!

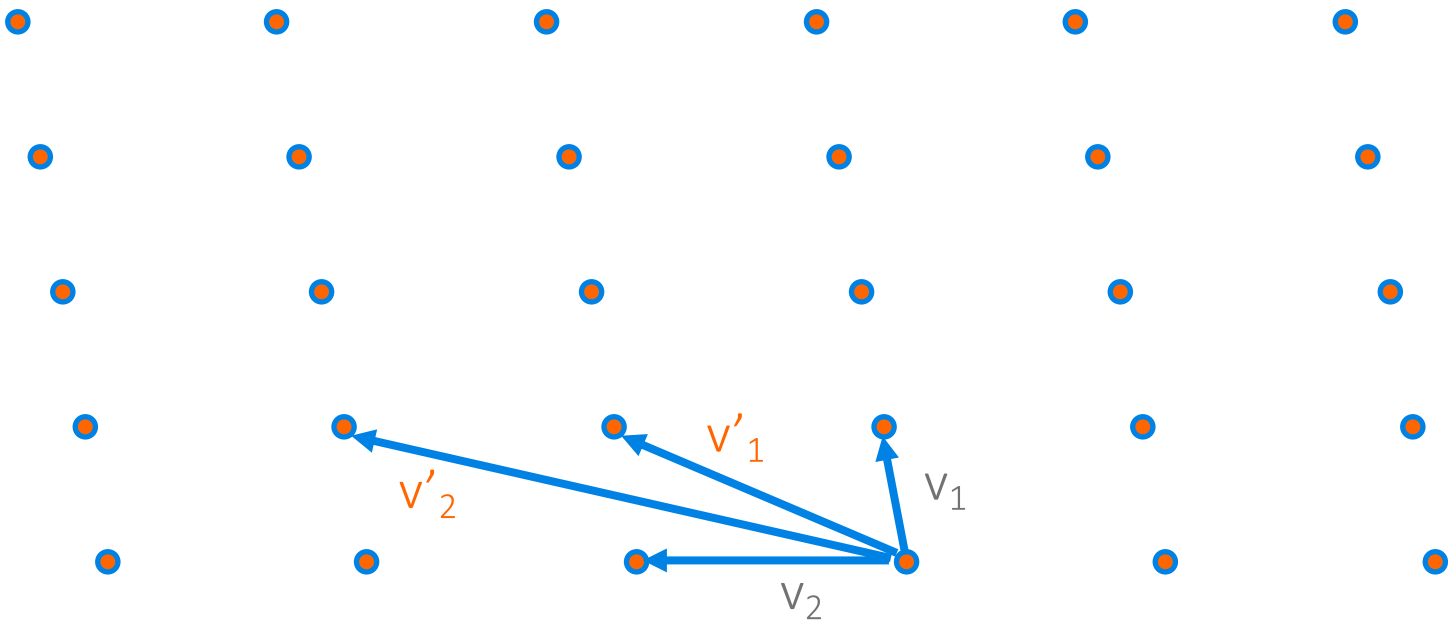
Preimage Sampleable Trapdoor Functions!

Generate (x, y) in two equivalent ways



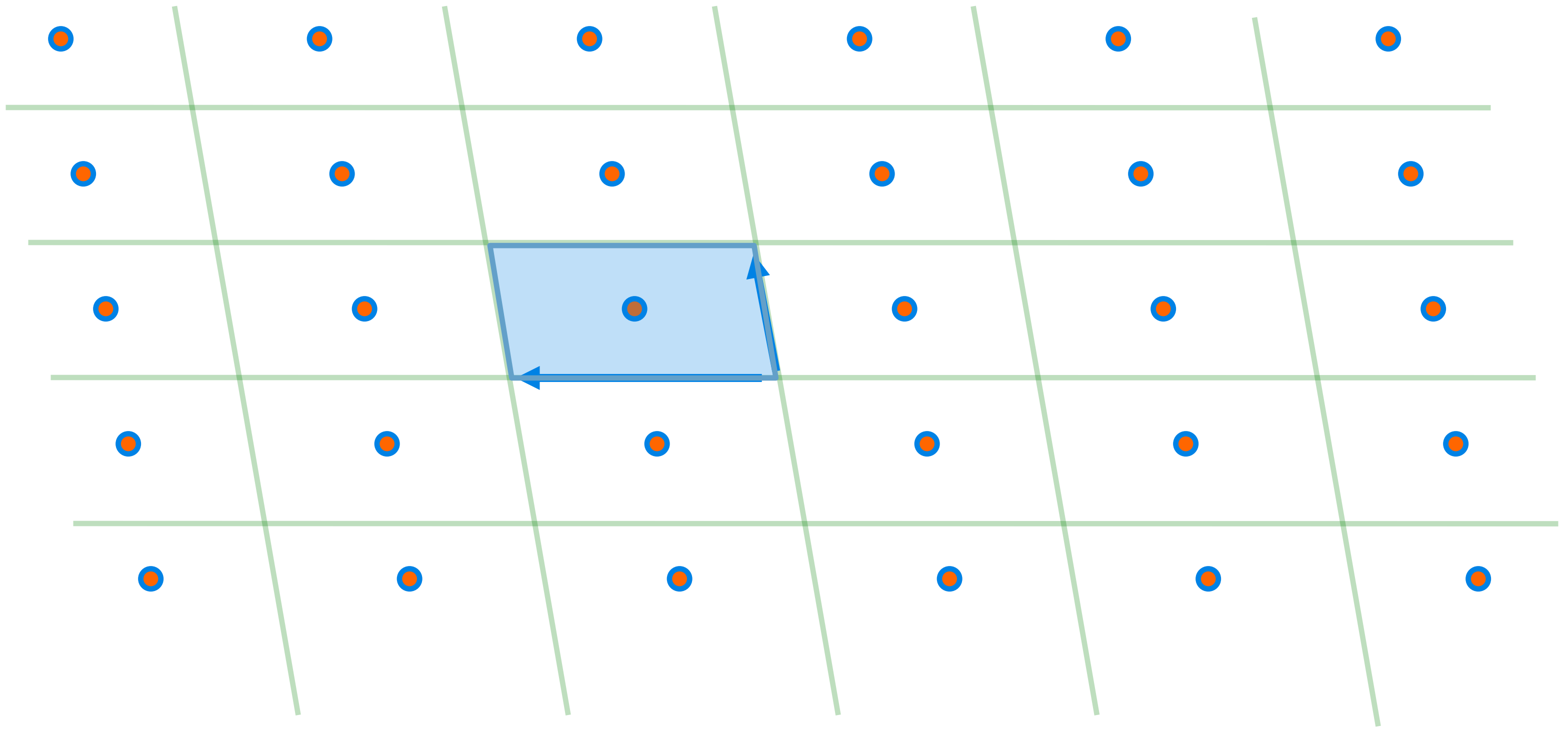
Same Distribution (Discrete Gaussian, Uniform) !

Lattice Trapdoors: Geometric View

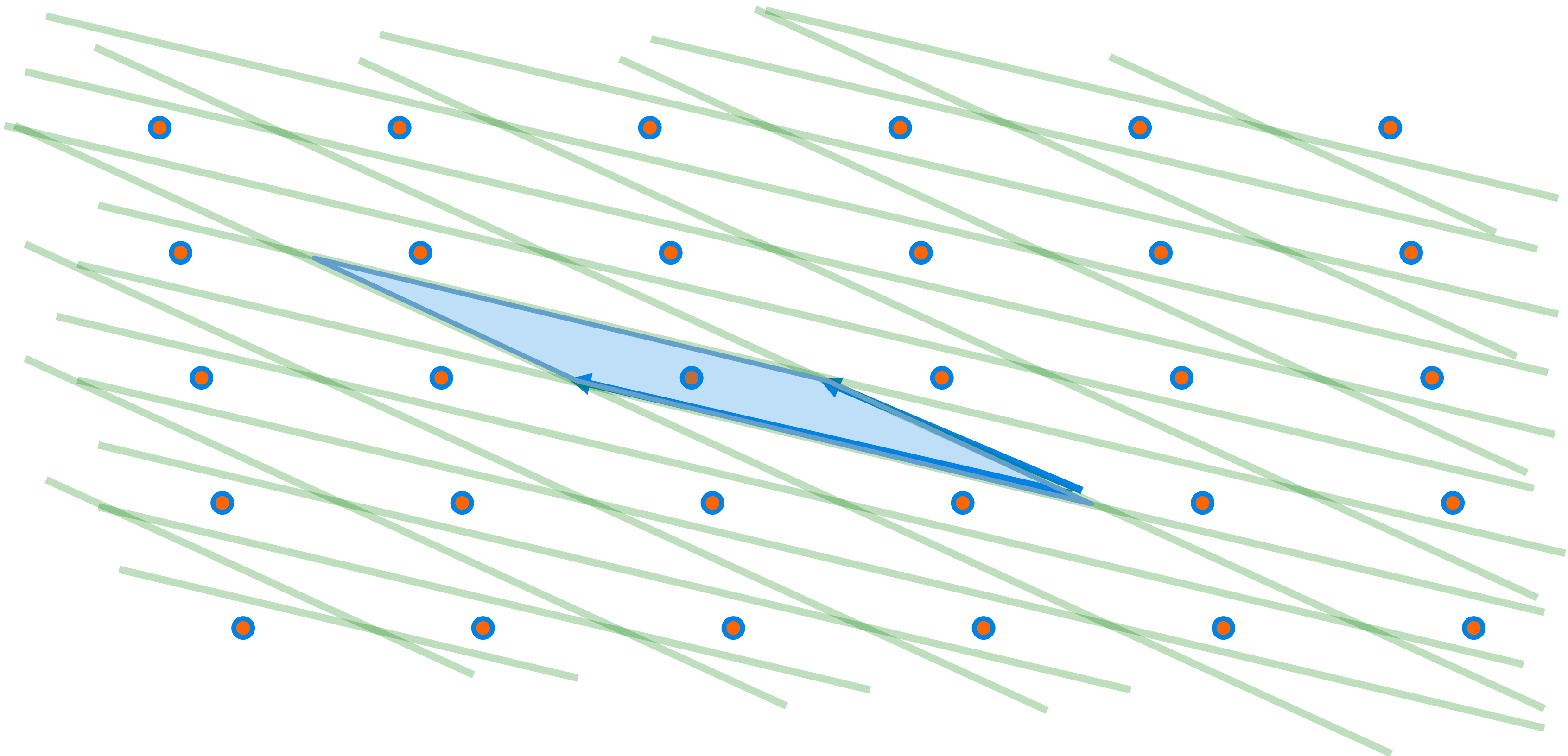


Multiple Bases

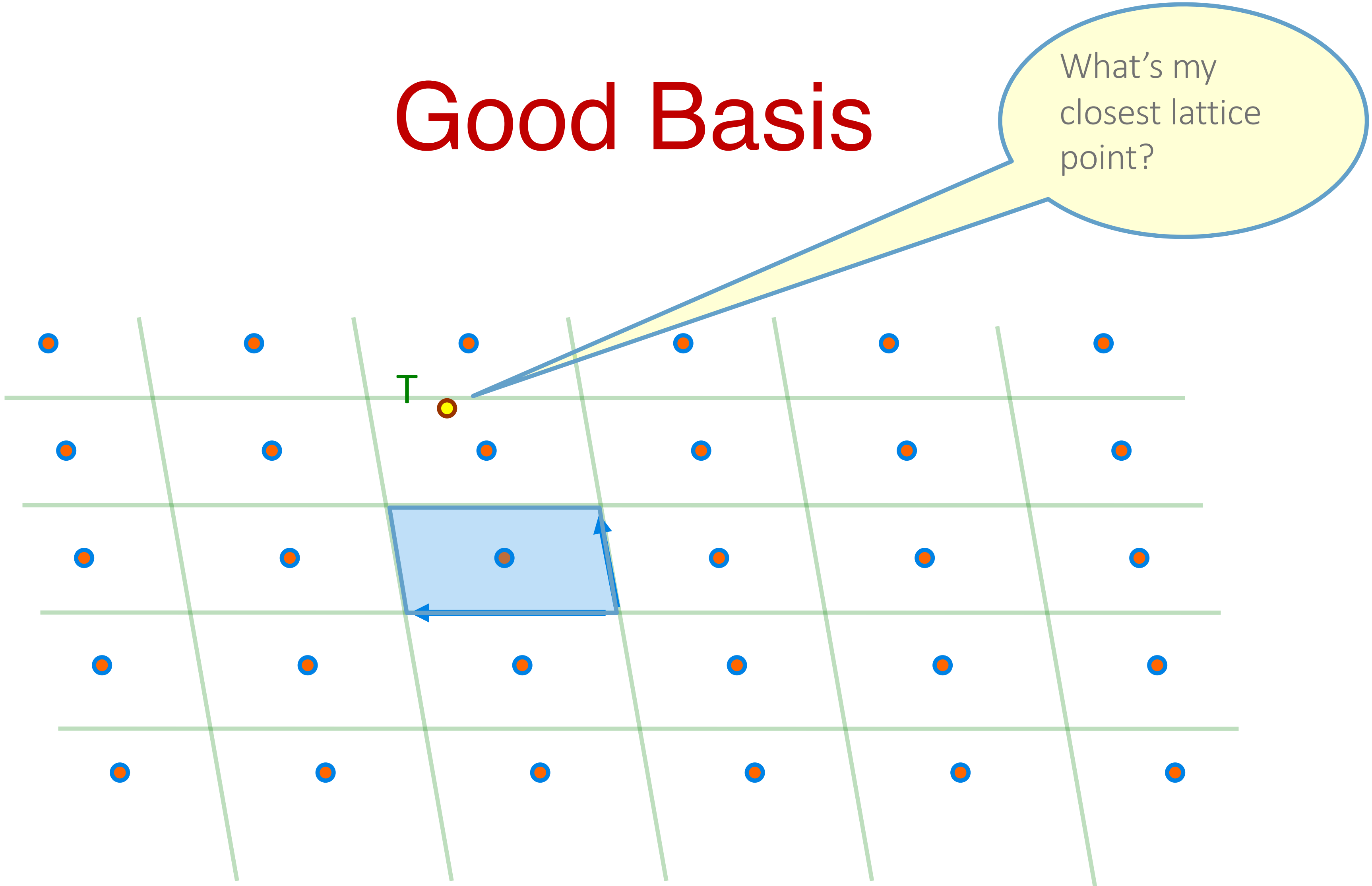
Parallelopipeds



Parallelopipeds

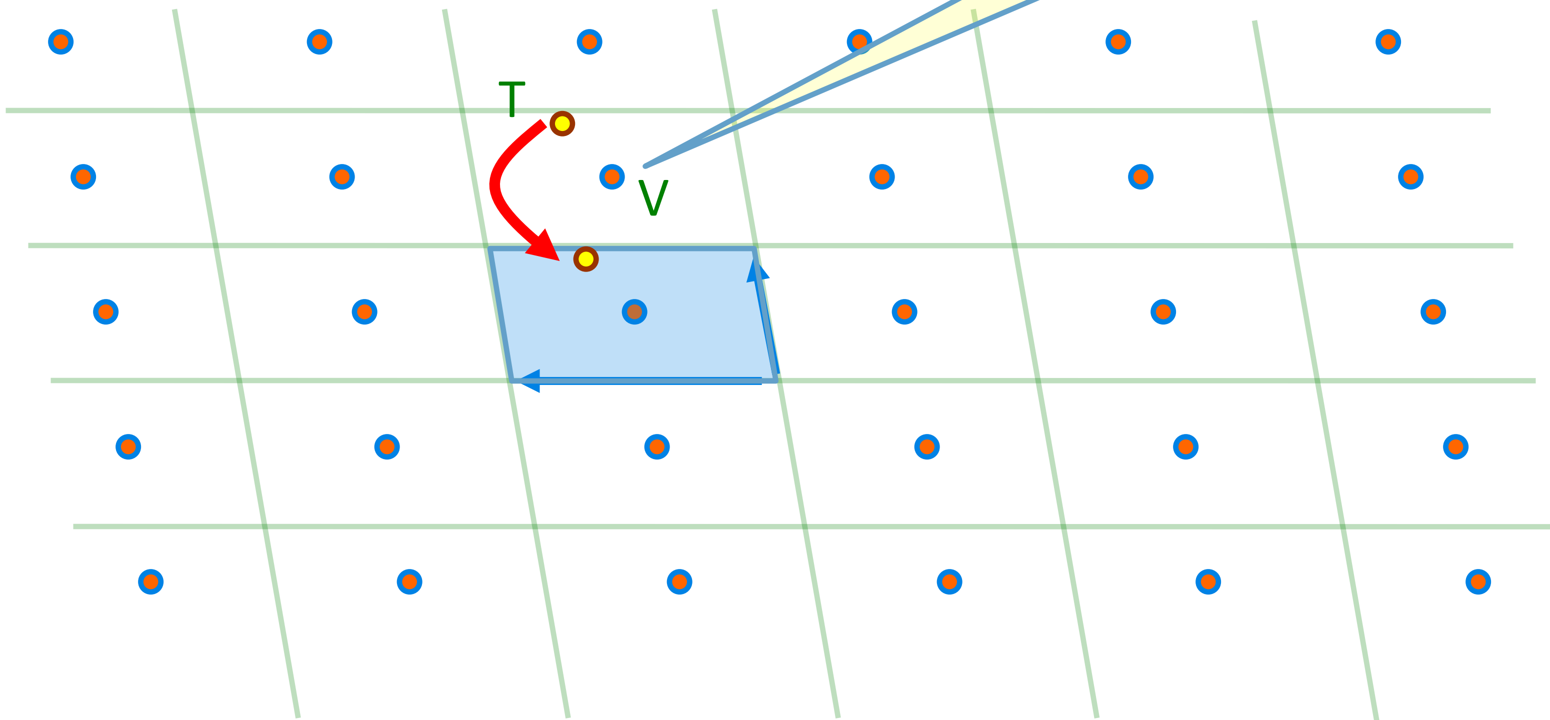


Good Basis



"Quite short" and "nearly orthogonal"

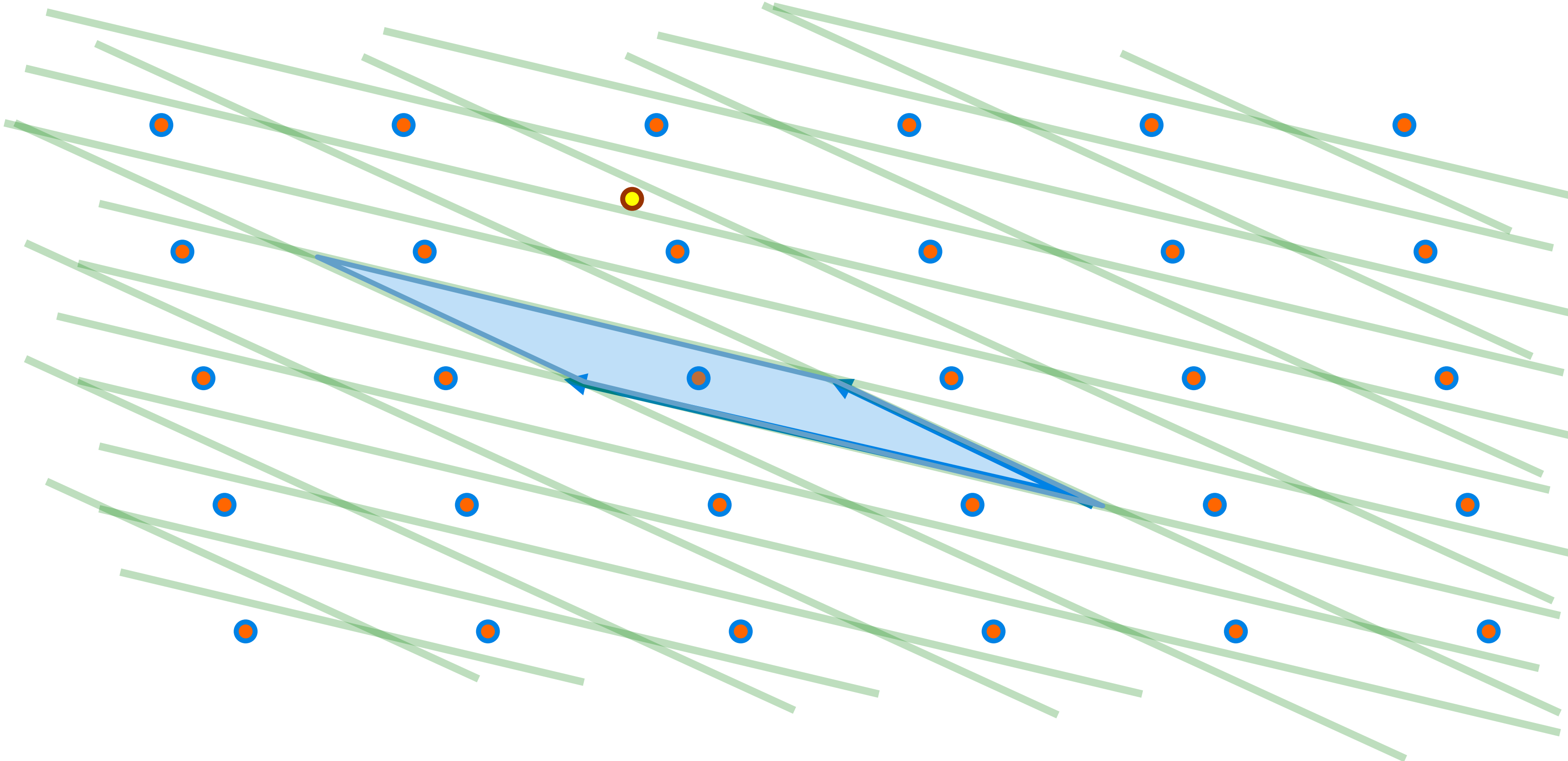
Good Basis



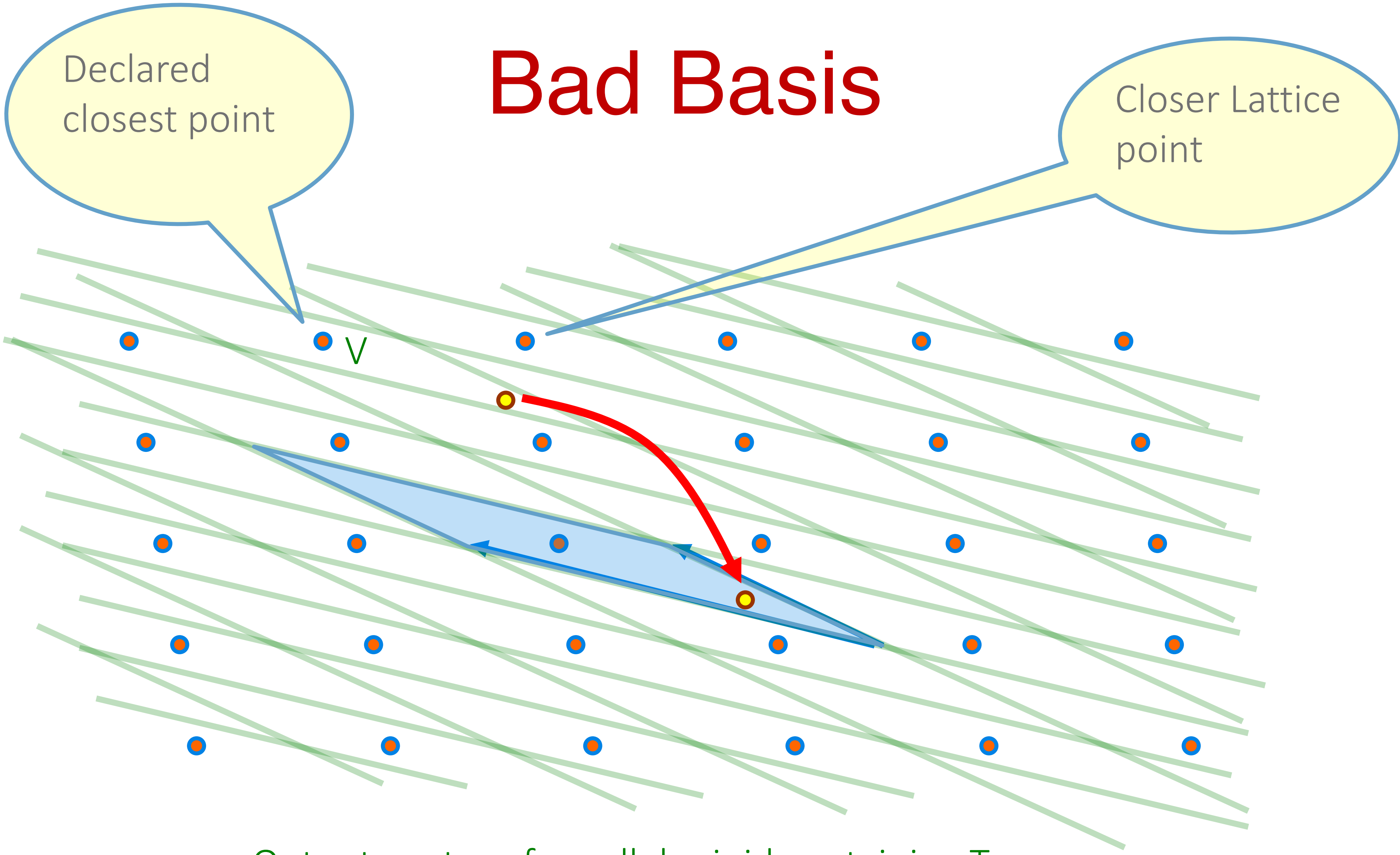
Output center of parallelopipid containing T

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

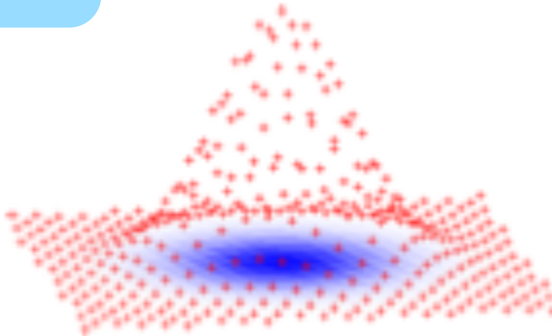
Inverting Our Function

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

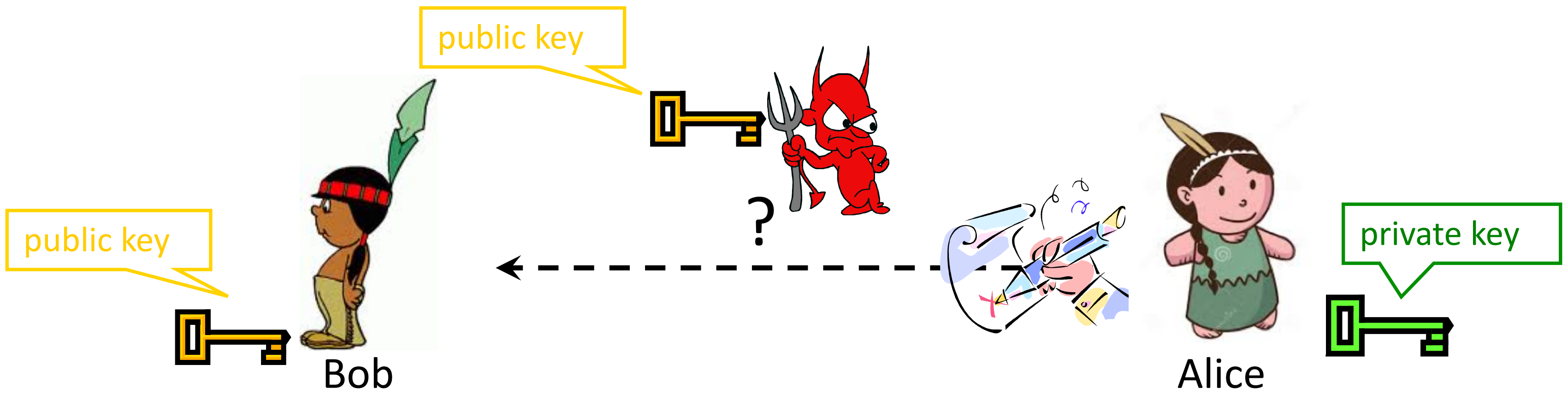


The Lattice

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

Short basis for $\boldsymbol{\Lambda}$ 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

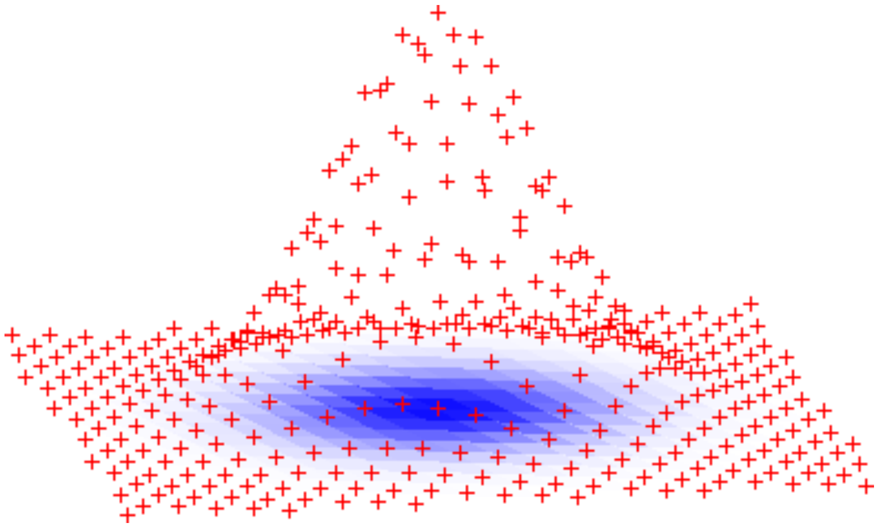
<u>Goal</u>: Alice sends a "digitally signed" message

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' $sk = \mathbf{T}$.
- 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!

Summary

- Post Quantum Crypto: Applications
- Basics of Lattices
- Hard Problems on Lattices
- Public Key Encryption
- Lattice Trapdoors
- Digital Signatures

Thank You

Images Credit: Hans Hoffman

Slides Credit: Daniele Micciancio, Chris Peikert