COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Final Details

About same length as midterm

Pick any **72 hour** period during the dates **May 17 – May 22**

- Don’t look at the final before your 72 hour period
- Email us when you first download the exam

Individual, but open notes/slides/internet…

Office Hours

No more Monday OH

During reading period/finals, OH will be by appointment

Today

CCA-secure PKE without random oracles

Secret sharing

Beyond COS433

Injective Trapdoor Functions

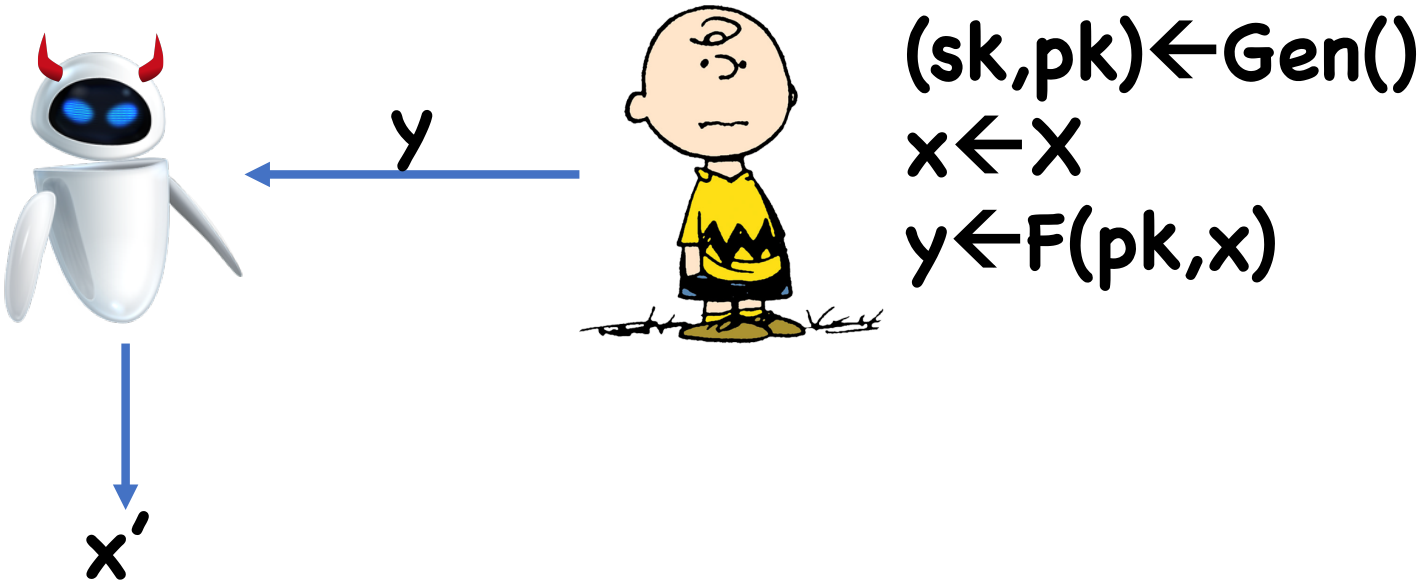
Domain **X**, range **Y**

Gen(): outputs **(pk,sk)**
F(pk,x∈X) = y∈Y, deterministic
F^{-1}(sk,y) = x

Correctness:
Pr[F^{-1}(sk, F(pk, x)) = x : (pk,sk)←Gen()] = 1

Correctness implies **F** is injective

Trapdoor Permutation Security



Injective TDFs from DH

Notation:

Let $A \in \mathbb{Z}_p^{n \times n}$

$g^A \in G^{n \times n}$, $(g^A)_{i,j} := g^{A_{i,j}}$

Let $H \in G^{n \times n}$, $v \in \mathbb{Z}_p^{n}$

$H^v \in G^n$, $(H^v)_i := \prod_j H_{i,j}^{v_j}$

Note: $((g^A)^v)_i = \prod_j g^{A_{i,j} v_j} = g^{(A \cdot v)_i}$, so $(g^A)^v = g^{A \cdot v}$

Injective TDFs from DH

Notation:

Let $\mathbf{h} \in G^n$, $A \in \mathbb{Z}_q^{n \times n}$

$^A\mathbf{h} \in G^n$, $(^A\mathbf{h})_i := \prod_j h_j^{A_{i,j}}$

$^A(g^v)$?

$(^A(g^v))_i = \prod_j g^{A_{i,j} v_j} = g^{(A \cdot v)_i}$, so $^A(g^v) = g^{A \cdot v}$

Injective TDFs from DH

Gen(): choose random $A \leftarrow \mathbb{Z}_q^{n \times n}$

$sk = A$, $pk = H = g^A$

$F(pk, x \in \{0,1\}^n) = H^x \ (= g^{A \cdot x})$

$F^{-1}(sk, h)$: $y \leftarrow {}^{A^{-1}}h \ (= g^{A^{-1} \cdot A \cdot x} = g^x)$

Then Dlog each component to recover x

Matrix DH Problems

Recall the DDH definition:

$$(g, g^a, g^b, g^{ab}) \approx_c (g, g^a, g^b, g^c)$$

Write as matrix:

$$g^{\begin{pmatrix} 1 & a \\ b & ab \end{pmatrix}} \approx_c g^{\begin{pmatrix} 1 & a \\ b & c \end{pmatrix}}$$

Matrix DH Problems

g^{A} vs g^{B}

$(A \leftarrow \mathbb{Z}_p^{n \times n})$ $(B$ random rank 1 matrix$)$

Theorem: If DDH holds, the matrix DH problem is hard for any n

Lossy Trapdoor Functions

$Gen_{inj}()$: outputs **(pk,sk)**

$Gen_{los}()$: outputs **pk**

$F(pk, x \in X) = y \in Y$, deterministic

$F^{-1}(sk, y) = x$

Correctness:

$Pr[\ F^{-1}(sk,\ F(pk,\ x)) = x : (pk,sk) \leftarrow Gen_{inj}()\ ] = 1$

If **$pk \leftarrow Gen_{los}()$**, then **$F(pk, \cdot\ )$** is "lossy"

Lossy Trapdoor Functions

Security: injective and lossy public keys are indistinguishable

Precisely:

$$\mathbf{pk: (pk,sk)\leftarrow Gen_{inj}()} \quad \approx_c \quad \mathbf{pk: pk\leftarrow Gen_{los}()}$$

Lossy TDFs from DH

$\mathbf{Gen_{inj}()}$: choose random $\mathbf{A \leftarrow \mathbb{Z}_q^{n \times n}}$

$\mathbf{sk = A,\ pk = H = g^A}$

$\mathbf{Gen_{los}()}$: Choose random rank-$\mathbf{1}$ $\mathbf{A \in \mathbb{Z}_q^{n \times n}}$

$\mathbf{pk = H = g^A}$

$\mathbf{F(pk,\ x \in \{0,1\}^n) = H^x\ (= g^{A \cdot x})}$

$\mathbf{F^{-1}(sk,\ h)}$: $\mathbf{y \leftarrow A^{-1}h\ (= g^{A^{-1} \cdot A \cdot x} = g^x\ )}$

Then Dlog each component to recover $\mathbf{x}$

Theorem: DDH → Matrix DDH → security

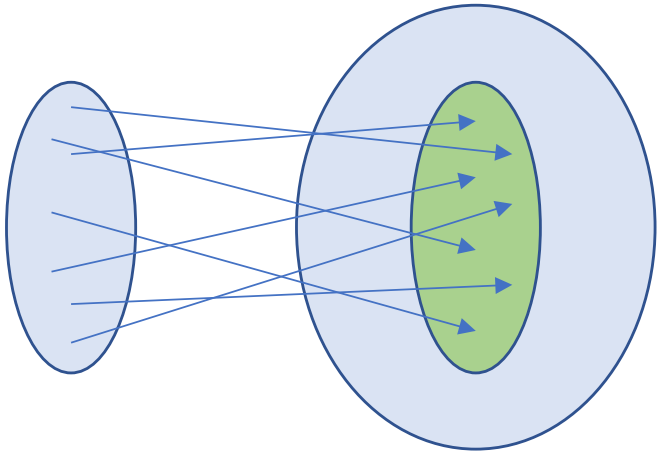
Lossy Functions

Suppose $H = g^A$ for rank-1 matrix A

What is the image of $F(pk, x) = H^x$?

Lossy Trapdoor Functions

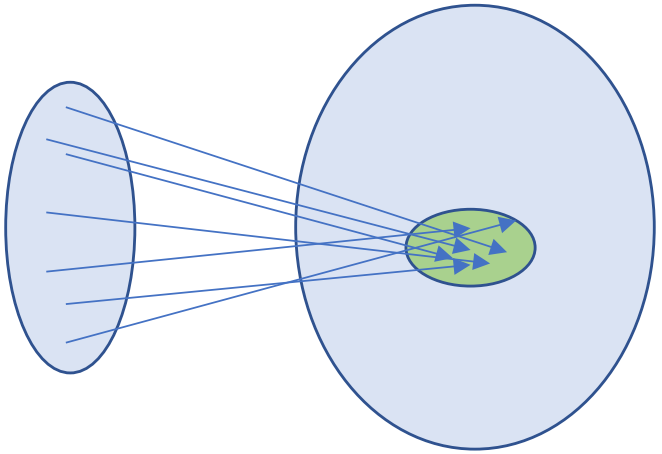
Injective Mode



$F(pk, \cdot)$ where
$(pk,sk) \leftarrow Gen_{inj}()$

$\approx_c$

Lossy Mode



$F(pk, \cdot)$ where
$pk \leftarrow Gen_{los}()$

Lossy Trapdoor Functions

LTDFs are also Injective TDFs

- In injective mode, there exists a unique pre-image for any image point
- In lossy mode, many collisions, so given **F(pk, x)**, impossible to find **x**
- Therefore, if possible to invert, then possible to distinguish injective from lossy:
  - Sample random **x**
  - Run inverter on **y = F(pk,x)**
  - Check if output **x′ = x**

CPA-Secure PKE from Inj. TDFs

Let **h** be a hardcore bit for the one-way function
x→F(pk,x)

Enc(pk,b) = F(pk,r), h(r)⊕b

Constructing Inj. TDFs with hardcore bits?

- **F'(pk, (r,x)) = (r , F(pk,x))**
- **h(r,x) = r⊕b**

CPA-Secure PKE from LTDFs

Can actually encrypt many bits at once

Ingredients:

- LTDF $(\mathbf{Gen_{inj}}, \mathbf{Gen_{los}}, \mathbf{G}, \mathbf{G^{-1}})$
- Pairwise independent hash function family $\mathbf{H}$

$$(\ |\text{Co-domain}(\mathbf{H})| \ll |\text{lossy range}|\ )$$

CPA-Secure PKE from LTDFs

Gen():

$(pk,sk) \leftarrow Gen_{inj}()$

$h \leftarrow H$

Enc((pk,h),m):

$r \leftarrow X$

$c_0 \leftarrow F(pk,r)$

$c_1 \leftarrow h(r) \oplus m$

Output (c_0,c_1)

Min-entropy

Definition: Given a distribution **D** over a set **X**, the min-entropy of **D**, denoted $H_\infty(D)$, is

$$-\min_x \log_2(\Pr[x \leftarrow D])$$

Examples:

- $H_\infty(\{0,1\}^n) = n$
- $H_\infty($ random n bit string with parity $0) = n-1$
- $H_\infty($ random $i>0$ where $\Pr[i] = 2^{-i}) = 1$

Leftover Hash Lemma

Lemma: Let **D** be a distribution on **X**, and **F** a family of pairwise independent functions from **X** to **Y**. Then

$$\Delta((f, f(D)) , (f, R)) \leq \varepsilon$$ where

- $f \leftarrow F$
- $R \leftarrow Y$
- $\log |Y| \leq H_\infty(D) + 2 \log \varepsilon$

CPA-Secure PKE from LTDFs

Security:

- First switch to lossy mode
- Ctxt has form $(c_0=F(pk,r),\ c_1=h(r)\oplus m)$
- Since lossy, even given c_0, r has min-entropy
- LHL: statistically close to $(c_0=F(pk,r),\ c_1=k\oplus m)$
- Now m completely hidden

All-But-One TDF

Gen(b∈B): outputs **(pk,sk)**
G(pk,b′,x∈X) = y∈Y, deterministic
G^{-1}(sk,b′,y) = x

Correctness: ∀**b′≠b**,
Pr[G^{-1}(sk,b′,G(pk,b′,x))=x:(pk,sk)←Gen(b)] = 1

If **pk←Gen(b)**, then **G(pk, b, ·)** has "very small" range

All-But-One TDF

Branch **b** is hidden:

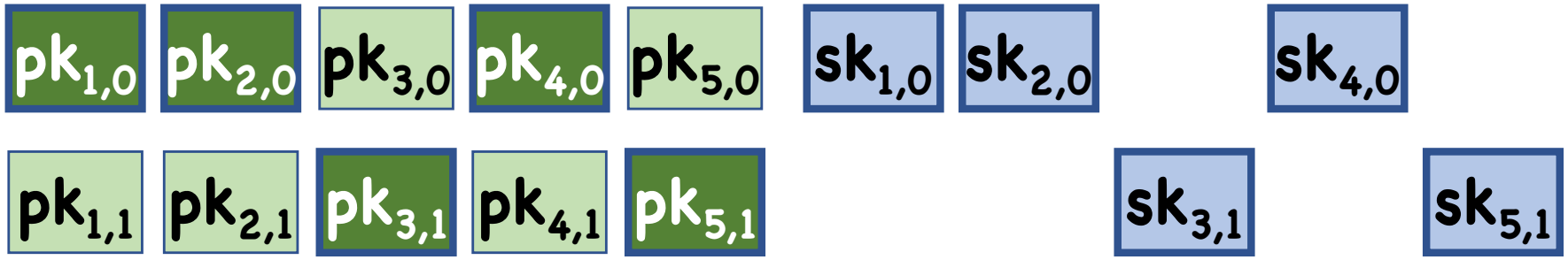
$\forall b_0, b_1,$

$\mathbf{pk: (pk,sk) \leftarrow Gen(b_0)} \quad \approx_c \quad \mathbf{pk: (pk,sk) \leftarrow Gen(b_1)}$

ABO from Lossy TDF

Suppose $\mathbf{B = \{0,1\}^n}$

Gen(b):

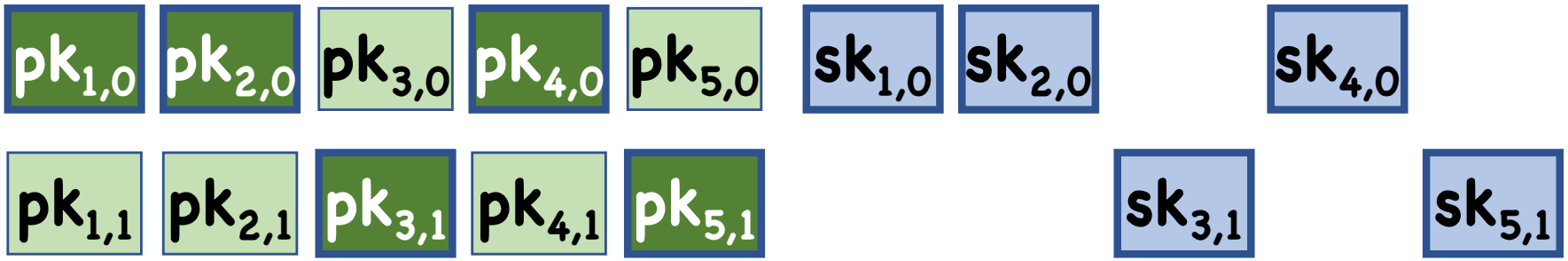


$(pk_{i,b_i}, sk_{i,b_i}) \leftarrow Gen_{inj}()$

$pk_{i,1-b_i} \leftarrow Gen_{los}()$

ABO from Lossy TDF

Gen(b):



$G(pk, b', x)$: $\left(\quad F(pk_{i,b_i'}, x) \quad \right)_{i=1,\ldots,n}$

$G^{-1}(sk, b', x)$: Use sk_{i,b_i} where $b_i \neq b_i'$

CCA-Secure PKE from ABO TDF

Ingredients:

- ABO TDF $\mathbf{(Gen_{ABO}, G, G^{-1})}$
- Strongly secure 1-time signature $\mathbf{(Gen_{sig}, Sign, Ver)}$
- Pairwise independent hash function family $\mathbf{H}$

CCA-Secure PKE from ABO TDFs

Gen():

$(pk,sk) \leftarrow Gen_{ABO}(b)$ for random b

$h \leftarrow H$

Enc((pk,h),m):

$r \leftarrow X$

$(vk,sk') \leftarrow Gen_{Sig}()$

$c_0 \leftarrow G(pk,vk,r)$

$c_1 \leftarrow h(r) \oplus m$

$\sigma \leftarrow Sign(sk', (c_0,c_1))$

Output (vk,c_0,c_1,σ)

Dec((sk,h), (vk,c_0,c_1,σ)):

Check $Ver(vk, (c_0,c_1), \sigma)$

$x' \leftarrow G^{-1}(sk,vk,c_0)$

Check $G(pk,vk,x') == c_0$

Output $h(x') \oplus c_1$

Theorem: If (Gen_{ABO}, G, G^{-1}) is a secure ABO TDF, $(Gen_{sig}, Sign, Ver)$ is a strongly secure 1-time signature scheme, and H is pairwise independent, then (Gen_{PKE}, Enc, Dec) is CCA-secure

Proof

Let $\mathbf{m_0^*, m_1^*}$ be challenger query, $\mathbf{(vk^*, c_0^*, c_1^*, \sigma^*)}$ be ctxt
Note that any CCA query must have $\mathbf{vk \neq vk^*}$

Hybrid 0: Encrypt $\mathbf{m_0^*}$

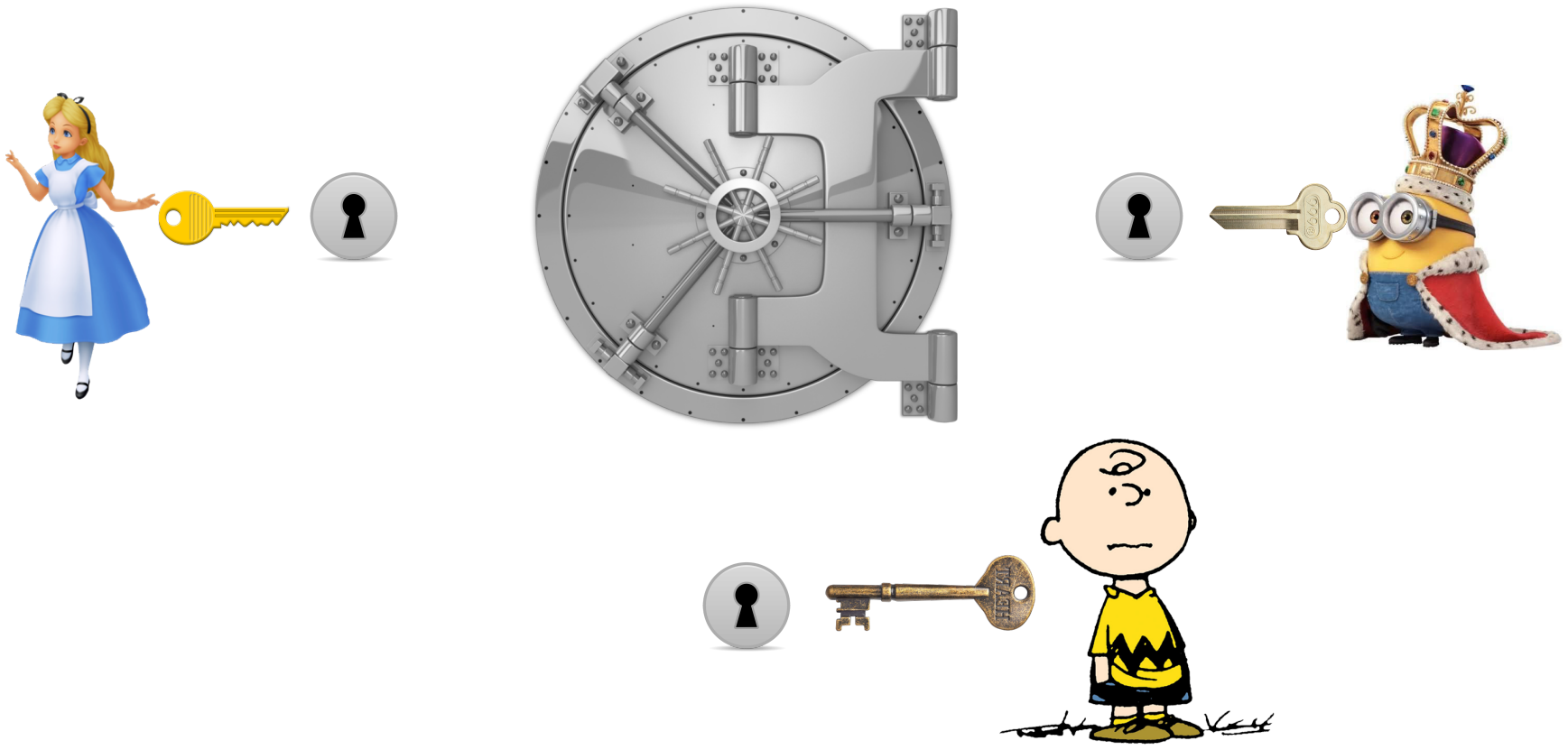
Hybrid 1: Change **pk** to **Gen(vk*)**
- Can choose $\mathbf{vk^*}$ at beginning
- Can still answer all CCA queries

Hybrid 2: Change $\mathbf{c_2^*}$ to $\mathbf{h(x^*) \oplus m_1^*}$
- Since lossy, $\mathbf{h(x^*)}$ is statistically close to random

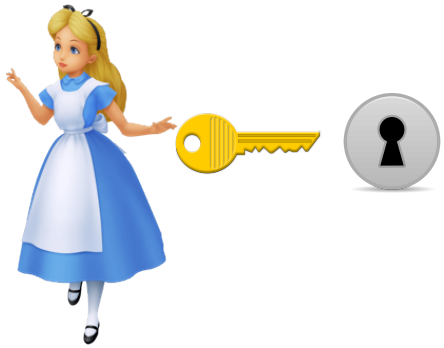
Hybrid 3: Change **pk** back to **Gen(b)**

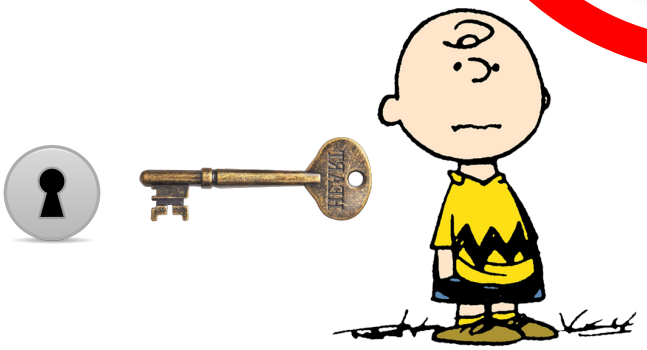
Secret Sharing

Vault should only open if both Alice and Bob are present



Vault should only open if Alice, Bob, and Charlie are all present





Vault should only open if any two of Alice, Bob, and Charlie are present

(Threshold) Secret Sharing

Syntax:

Share(k,t,n) outputs $(sh_1,\ldots,sh_n)$

Recon($(sh_i)_{i\in S}$ **)** outputs **k′**

Correctness: $\forall$ **S s.t. |S|≥t**

If $(sh_i)_{i=1,\ldots,n} \leftarrow$ **Share(k,t,n),** then

Pr[Recon($(sh_i)_{i\in S}$ **) = k] = 1**

(Threshold) Secret Sharing

Security:

For any S, $|S|<t$, given $(sh_i)_{i\in S}$, should be impossible to recover k

$$(sh_i)_{i\in S}: (sh_i)_{i=1,\dots,n} \leftarrow \mathbf{Share}(k_0,t,n)$$

$$\approx$$

$$(sh_i)_{i\in S}: (sh_i)_{i=1,\dots,n} \leftarrow \mathbf{Share}(k_1,t,n)$$

n-out-of-**n** Secret Sharing

Share secret **k** so that only can only reconstruct **k** if all **n** users get together

Ideas?

Shamir Secret Sharing

Let **p** be a prime **> n, ≥#(k)**

Share(k,t,n):

- Choose a random polynomial **P** of degree **t-1** where **P(0) = k**
- **sh_i = P(i)**

Recon($(sh_i)_{i \in S}$): use shares to interpolate **P,** then evaluate on **0**

Shamir Secret Sharing

Correctness:

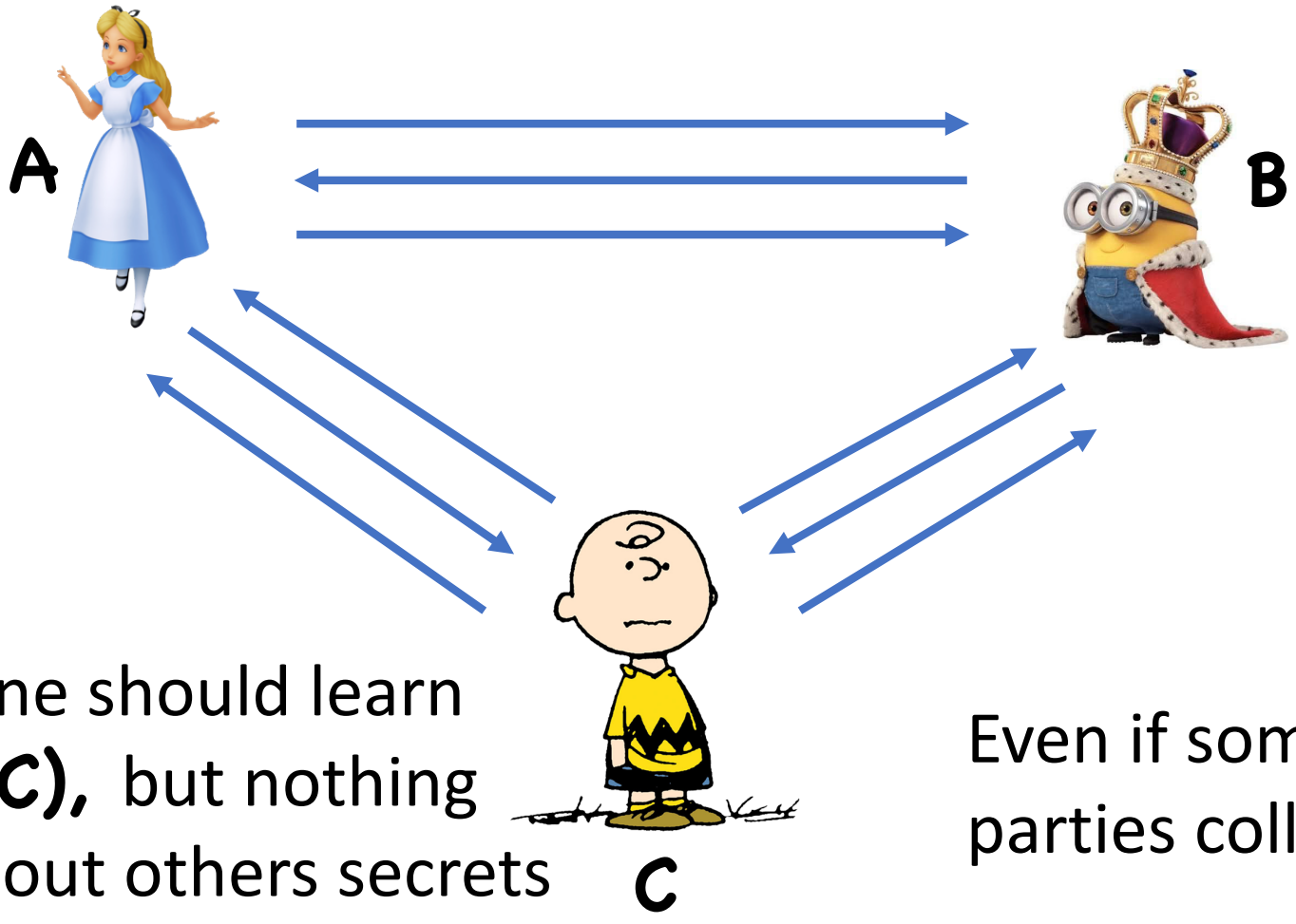
- **t** input/outputs (shares) are enough to interpolate a degree **t–1** polynomial

Security:

- Given just **t–1** inputs/outputs, **P(0)** is equally likely to be any value

Beyond COS 433

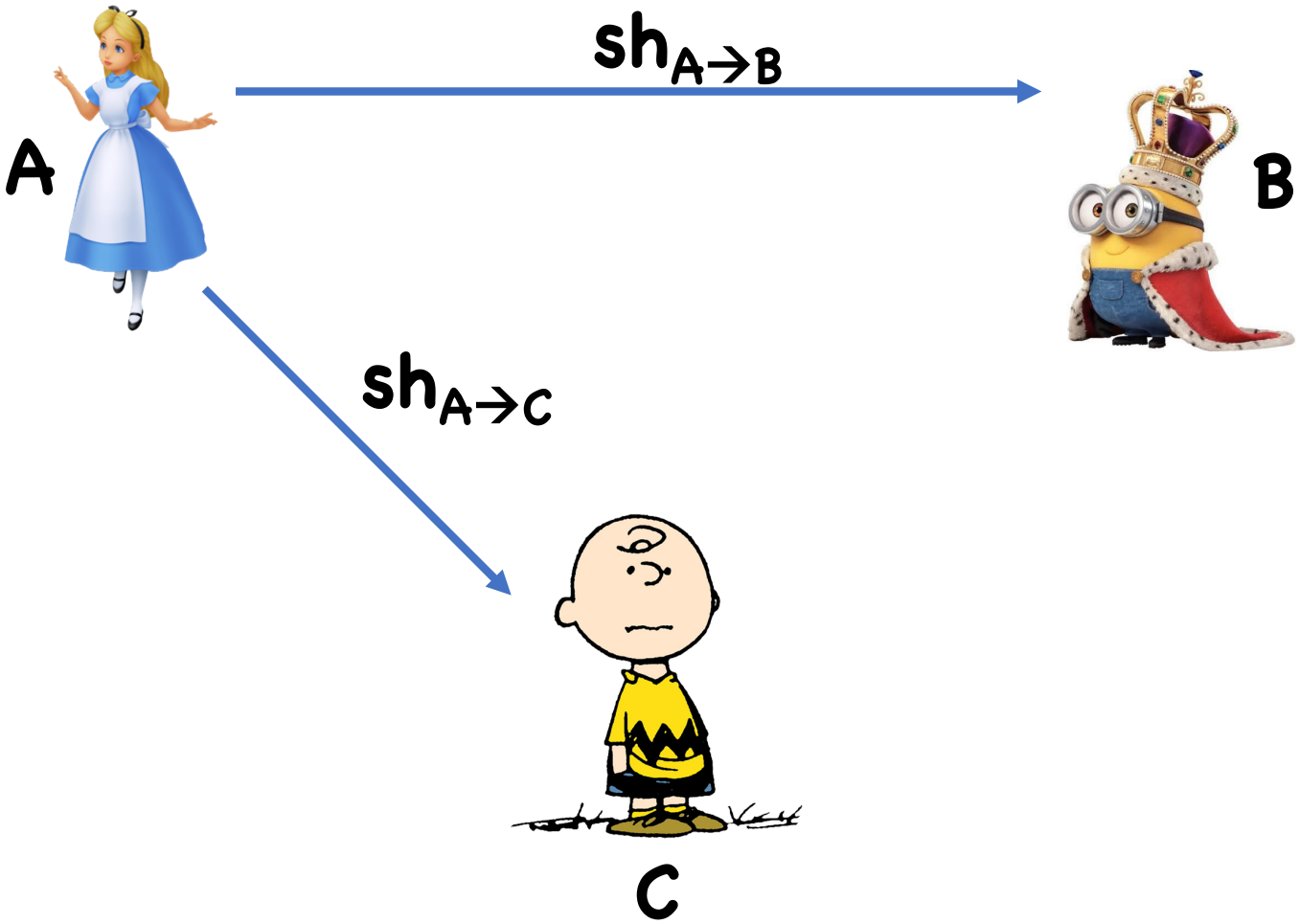
Multiparty Computation



Everyone should learn **F(A,B,C),** but nothing else about others secrets

Even if some parties collude!!

One Approach



One Approach

A

$sh_{A \to A}$
$sh_{B \to A}$
$sh_{C \to A}$

B

$sh_{A \to B}$
$sh_{B \to B}$
$sh_{C \to B}$

C

$sh_{A \to C}$
$sh_{B \to C}$
$sh_{C \to C}$

Additivity of Shamir SS

Suppose we have:

- Share $\mathbf{sh_i = P(i)}$ of secret **k**, and
- share $\mathbf{sh'_i = P'(i)}$ of secret **k′**

$\mathbf{sh_i + sh'_i}$ is a secret share of **k+k′**:

- $\mathbf{sh_i + sh'_i = P(i) + P'(i) = (P+P')(i)}$
- $\mathbf{(P+P')(0) = P(0) + P'(0) = k+k'}$

One Approach

Shamir SS is additive, so users can add shares together

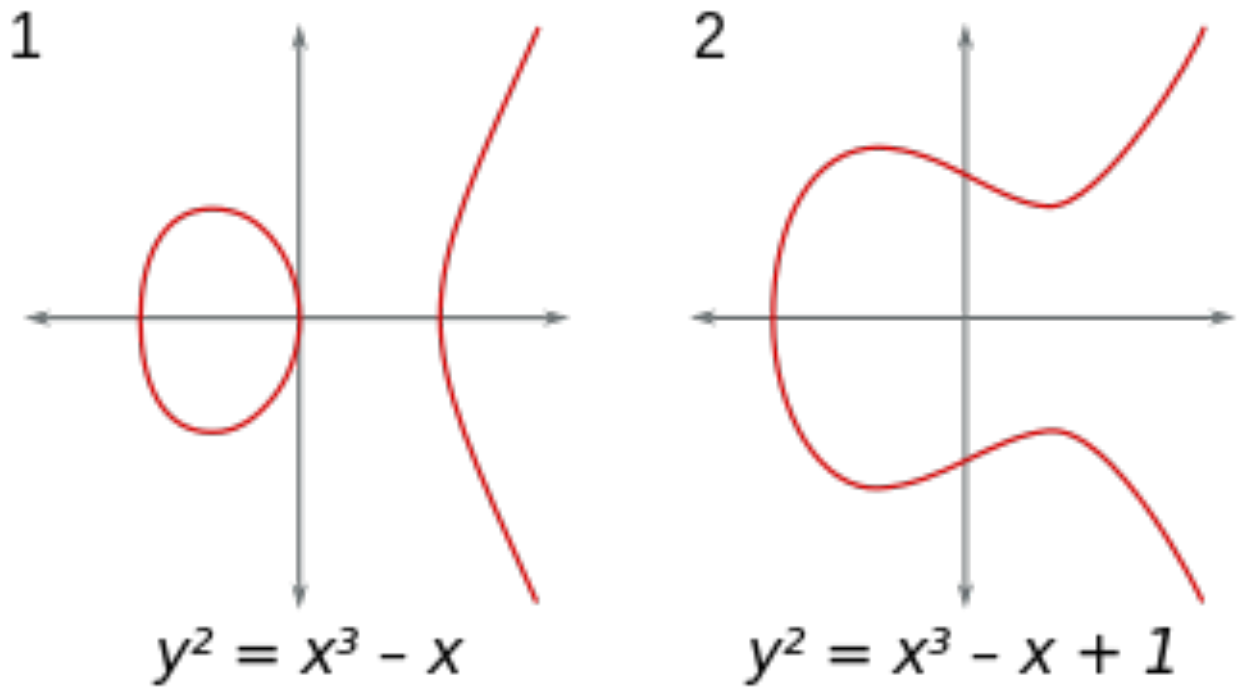
Cannot directly multiply shares, but possible with a little extra interaction

Therefore, can compute shares of arbitrary functions of inputs

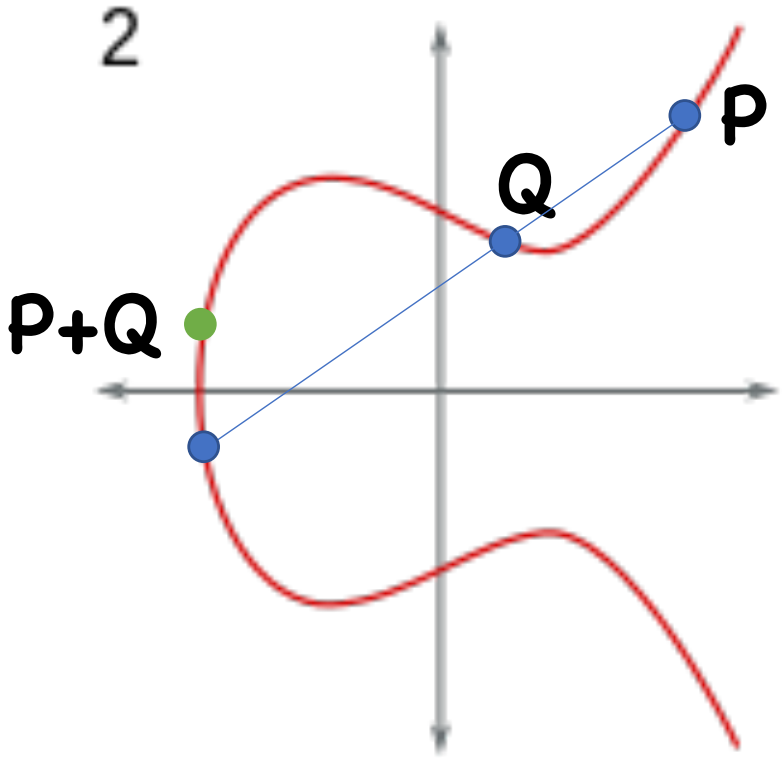
Finally, everyone exchanges shares to recover answer

Elliptic Curves

$$y^2 = a\,x^3 + b\,x^2 + c\,x + d$$



Group Law on ECs



ECs for Crypto

Consider EC over finite field

Set of solutions form a group

Dlog in group appears hard

- Given $aP = (P+P+\ldots+P)$, find a
- Can use in crypto applications

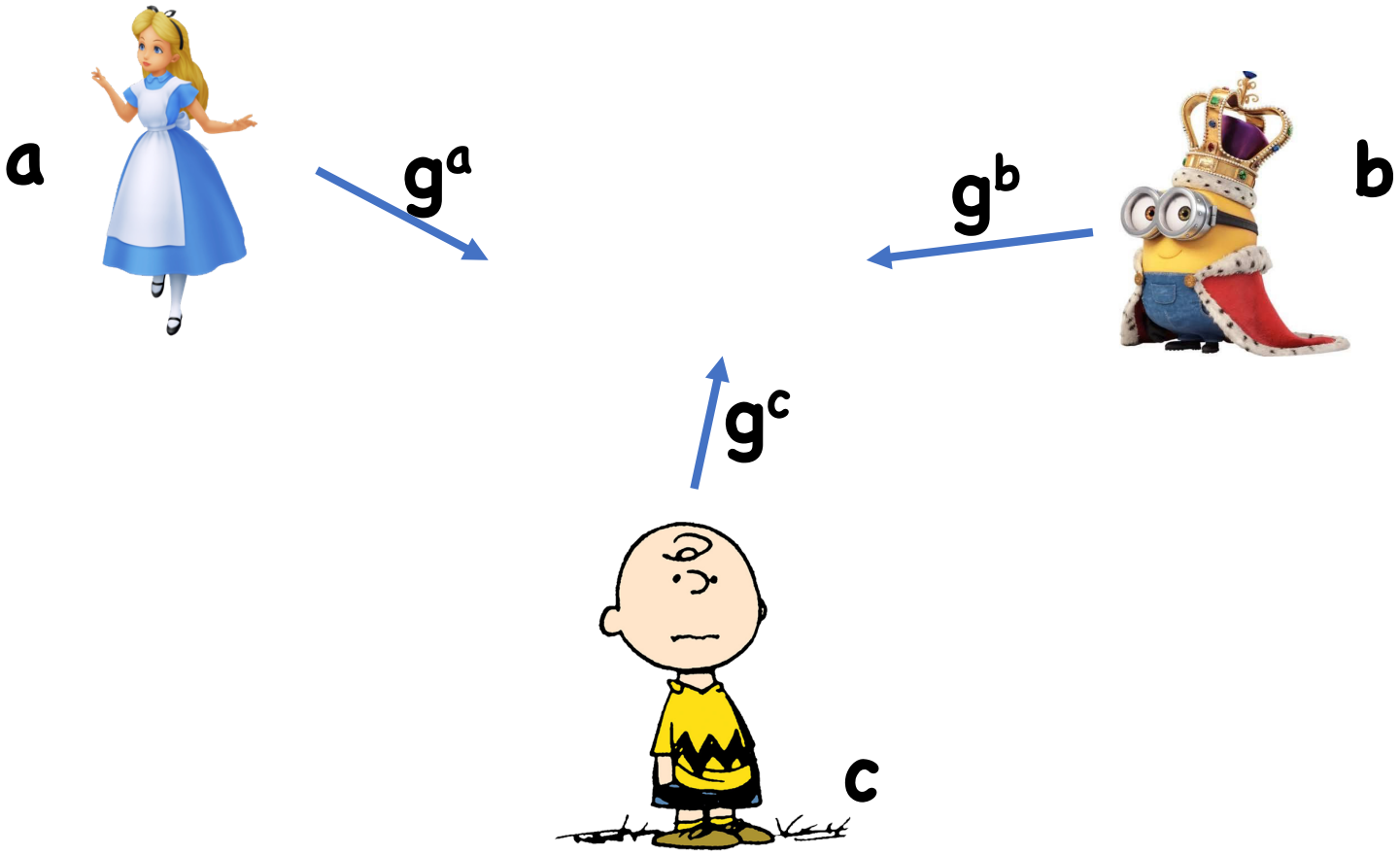
Bilinear Maps

On some Elliptic curves, additional useful structure

Map $e: G \times G \rightarrow G_2$

- $e(g^a, g^b) = e(g,g)^{ab}$

3-party Key Exchange



Shared key = $e(g,g)^{abc}$

Bilinear Maps

Extremely powerful tool, many applications beyond those in COS 433

- 3 party *non-interactive* key exchange
- Identity-based encryption
- Broadcast encryption

Multilinear Maps

Map $e: G_n \rightarrow G_2$

- $e(g^a, g^b, \ldots) = e(g,g,\ldots)^{ab\ldots}$

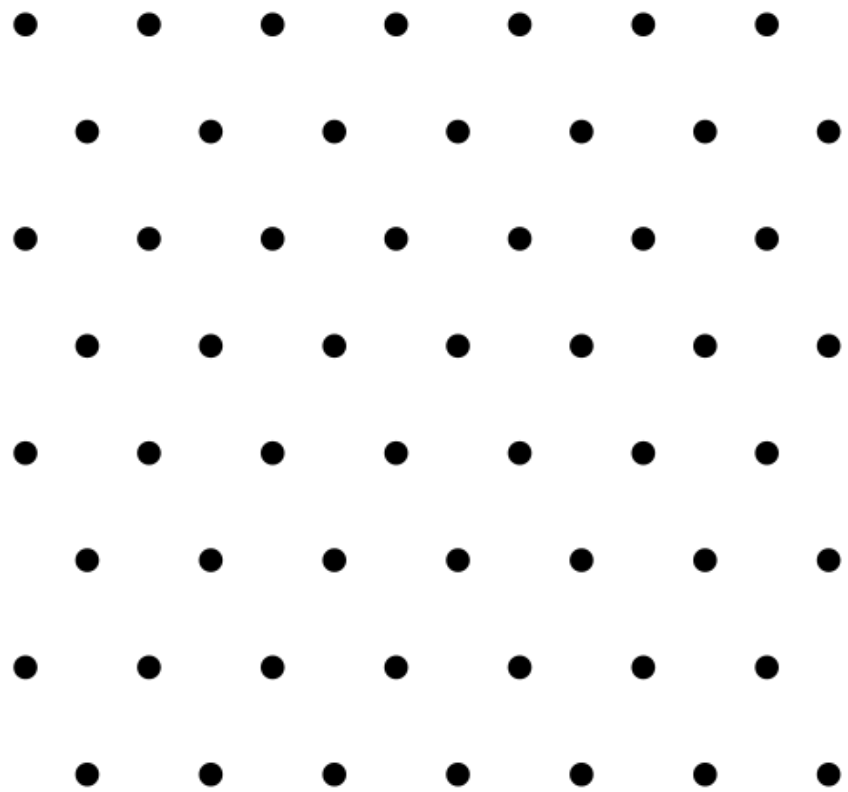
Many more applications that bilinear maps:

- n+1 party non-interactive key exchange
- Obfuscation
- …

Unfortunately, don't know how to construct from elliptic curves

- Recently, constructions based on other math

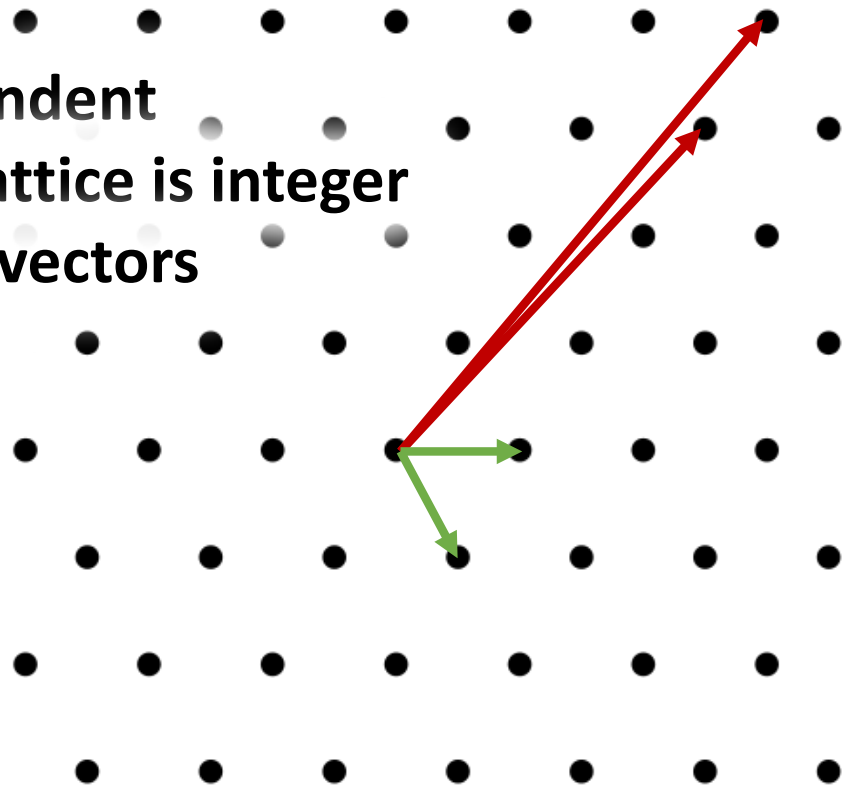
Lattices



Lattices

Basis:

- **Linearly independent**
- **Every point in lattice is integer combo of basis vectors**



Lattices

Hard problems in lattices:

- Given a basis, find the shortest vector in the lattice
- Given a basis an a point not in the lattice, find the closest lattice point

Can base much crypto on approximation versions of these problems

- Basically everything we've seen in COS433, then some

Fully Homomorphic Encryption

In homework, you saw additively/multiplicatively homomorphic encryption:

Enc(pk, x) + Enc(pk, y) = Enc(pk, x+y)

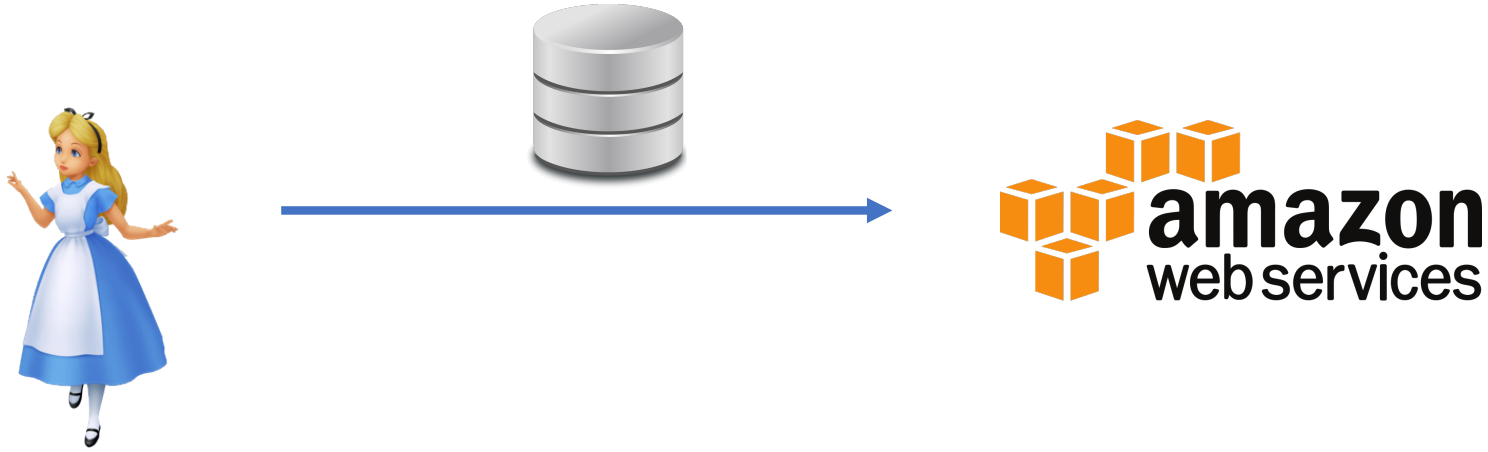
OR

Enc(pk, x) × Enc(pk, y) = Enc(pk, x×y)

What if you could do both simultaneously?

- Arbitrary computations on encrypted data

Delegation

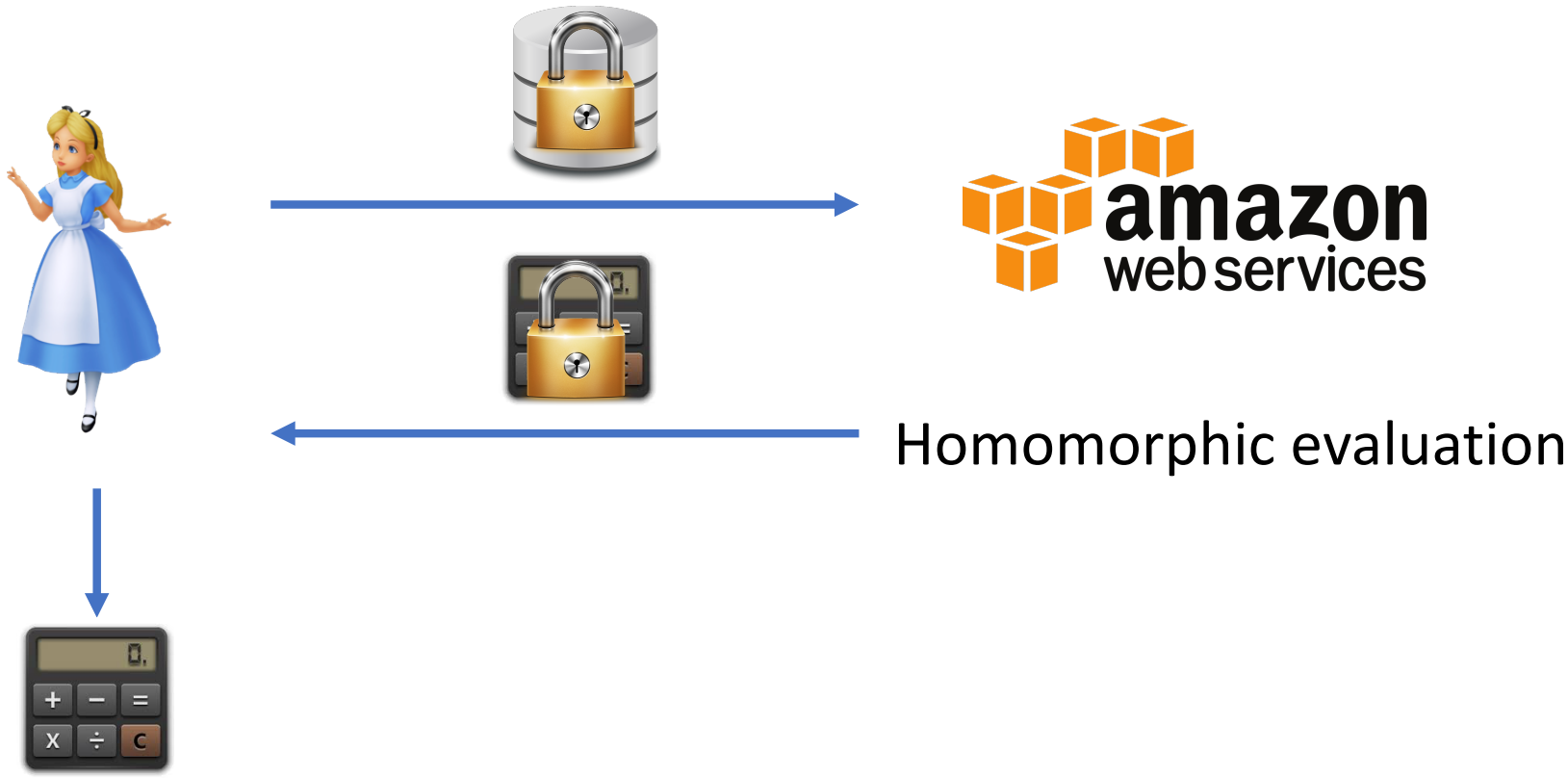


Doesn’t want Amazon to learn sensitive data

Delegation

Now, Alice wants Amazon to run expensive computation on data

Delegation



Quantum Computing

Computers that take advantage of quantum physics

Turns out, good at solving certain problems

- Dlog in any group ($\mathbb{Z}_p^*$, ECs)
- Factor integers

Also can speed up brute force search:

- Invert OWF in time $2^{n/2}$
- Find collisions in time $2^{n/3}$

Quantum Computing

To protect against quantum attacks, must:

- Must increase key size
  - 256 bits for one-way funcitons
  - 384 bits for collision resistance
- Must not use DDH/Factoring
  - Lattices instead

Quantum computers still at least a few years away, but coming

COS 533

Covers handful of topics in these areas

Class time: **MW 11am-12:20pm**