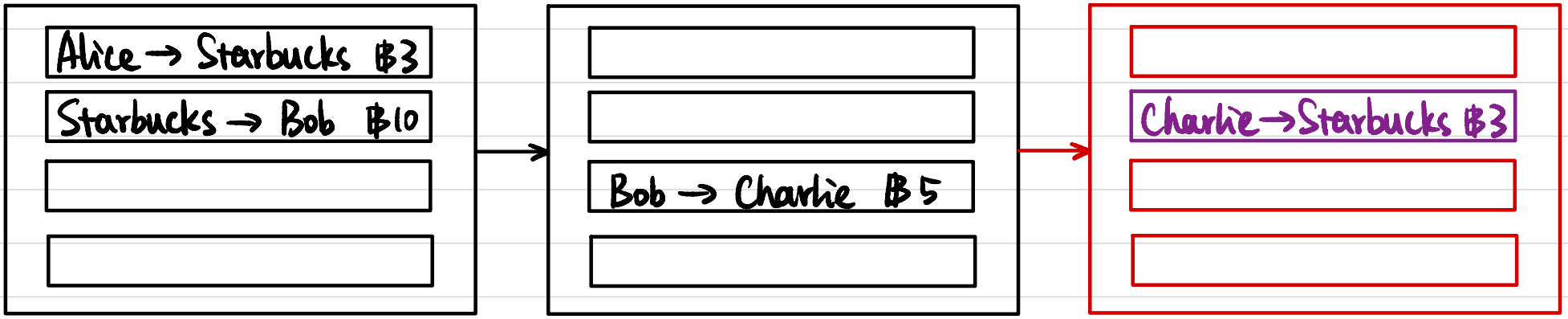
CSCI 1515 Applied Cryptography

This Lecture:

- Blockchain & Cryptocurrencies (Continued)
- Elliptic Curve Cryptography
- Introduction to Secure Multi-Party Computation

Blockchain



- Public ledger that everyone can view & verify
- Maintained by "miners" in a distributed way

Step 1: Charlie wants to make a transaction Charlie → Starbucks ฿3
↳ broadcasts it to the entire network

Step 2: All miners collect all transactions in the network
- Verify validity
  - ① initiated by sender ← Digital Signatures
  - ② enough balance in sender's account
- Agree on next block
  - How?

Step 3: Repeat

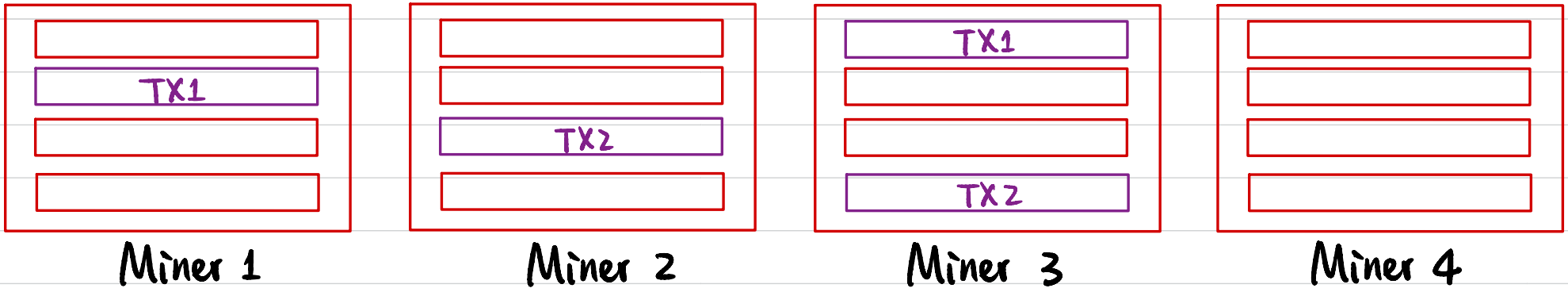
Consensus Protocol

TX1 = Charlie → Starbucks ฿3 :

$m_2 = (vk_C, vk_S, 3)$ $\sigma_2 \leftarrow Sign_{sk_C}(m_2)$

TX2 = Charlie → Alice ฿4 :

$m_3 = (vk_C, vk_A, 4)$ $\sigma_3 \leftarrow Sign_{sk_C}(m_3)$

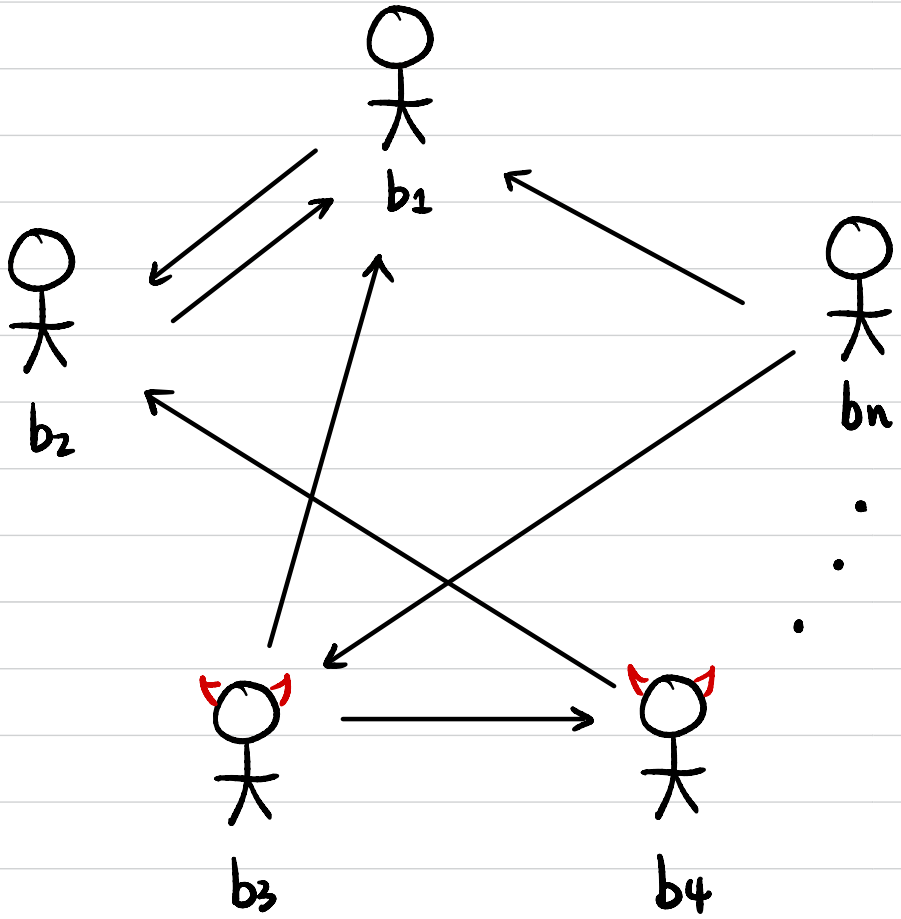


"permissionless"

WANT: ① All miners agree on the same block

② New block is valid

Byzantine Agreement



Agree on a block

Byzantine Fault Tolerance (BFT) Protocol:

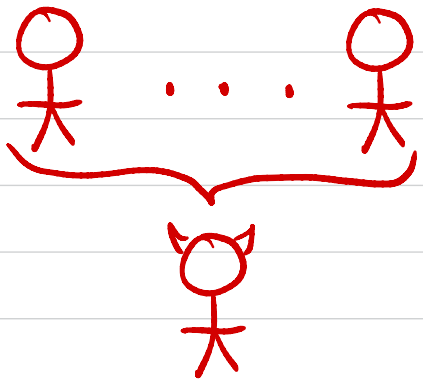
If $n \geq 3t+1$, (necessary)

then it's possible to reach consensus.

Assume $t < n/3$, then agree on a valid block.

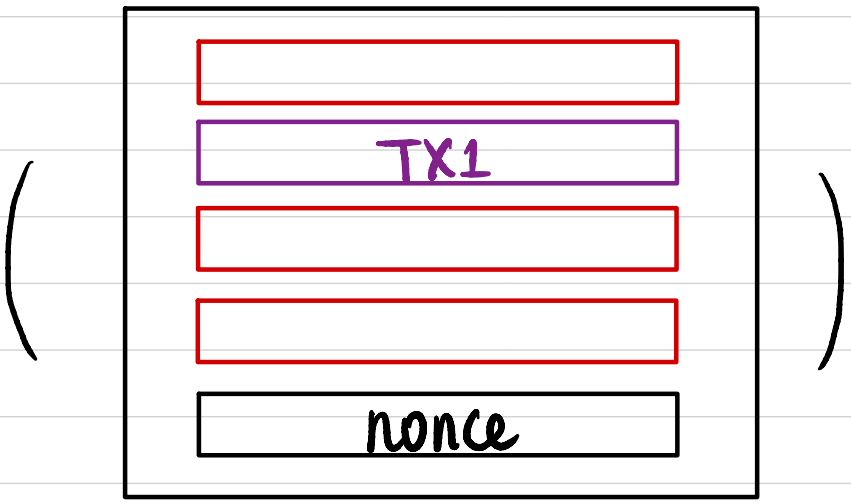
Any problem?

"Sybil Attack"



Proof of Work (PoW)

Miner 1:

Hash 

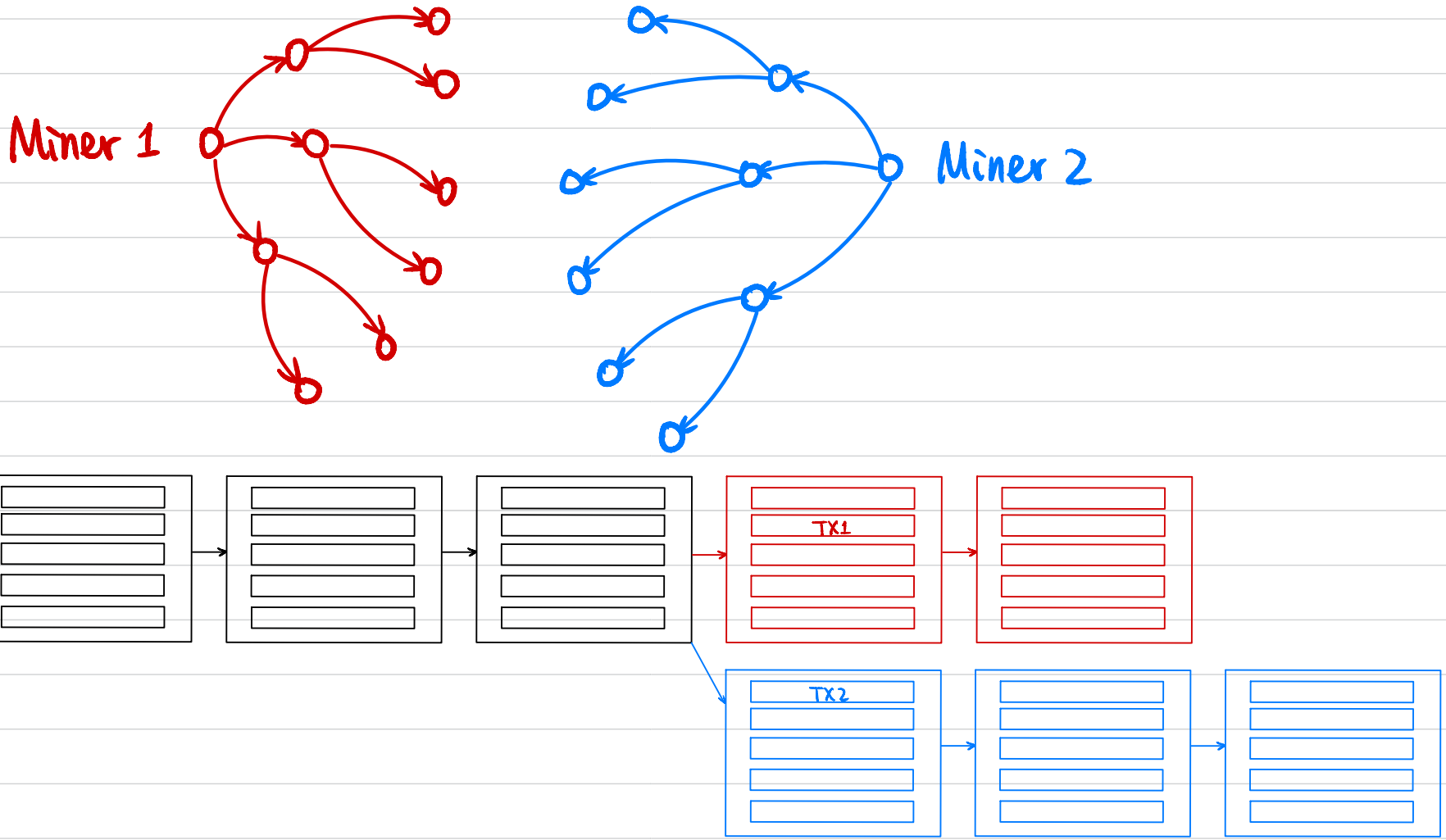
$= \underbrace{00\cdots0}_{30}1011\cdots0$

Find nonce s.t. Hash(block) has $\geq$ 30 leading 0's.

Consensus Protocol:

Whoever first finds a block that hashes to a value w/ $\geq$ 30 leading 0's, that block becomes the next block.

Proof of Work (PoW)



Longest Chain Rule: Always adopt the longest chain.

Assuming honest majority of computation power, the longest chain is always valid.

Blockchain

- Efficient verification of sufficient balance: Merkle Tree
- Settlement of a transaction:
  Included in a block which is ≥ 6 blocks deep (~ 1 hr)
- Dynamically adjust # leading 0's s.t. each block takes $\sim$ 10min to mine
  Last 1 hr: > 6 blocks: increase # leading 0's
  < 6 blocks: decrease # leading 0's
- Miners' motivation:
  - transaction fee
  - new coin generated in each block goes to miner
- Extensions
  - Fast verification (SNARGs)
  - Proof of Stake (PoS)
  - Anonymous transactions (zk-SNARGs)
  - Smart Contracts
  - Public Bulletin Board

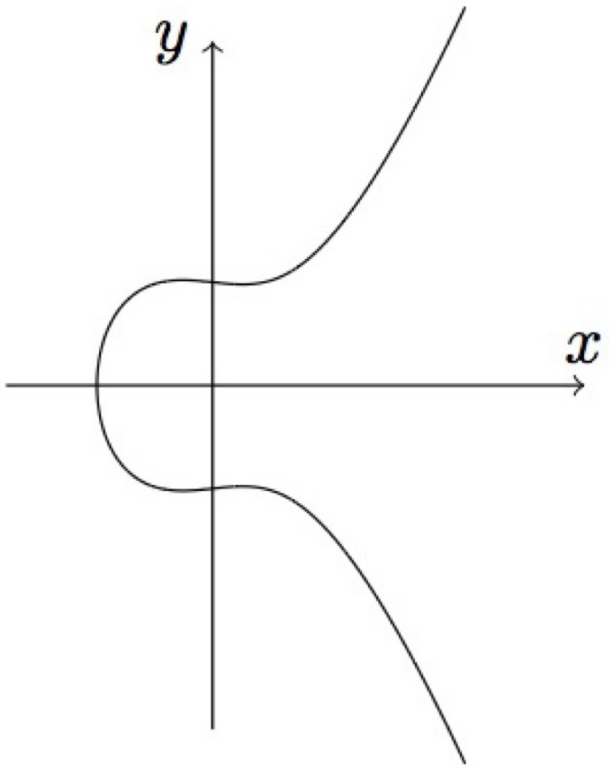
Elliptic Curve Cryptography

Cyclic group G of order q with generator g where DLOG/CDH/DDH holds.

How large is q? (128-bit security)

- Integer groups: $q \sim 2048$ bits
- Elliptic curve groups: $q \sim 256$ bits
  - Additional structure: bilinear pairings

Elliptic Curves



$$y^2 = x^3 + ax + b$$

$$(4a^3 + 27b^2 \neq 0)$$

Example: $y^2 = x^3 - x + 9$

points: $(0, \pm 3)$
$(1, \pm 3)$
$(-1, \pm 3)$

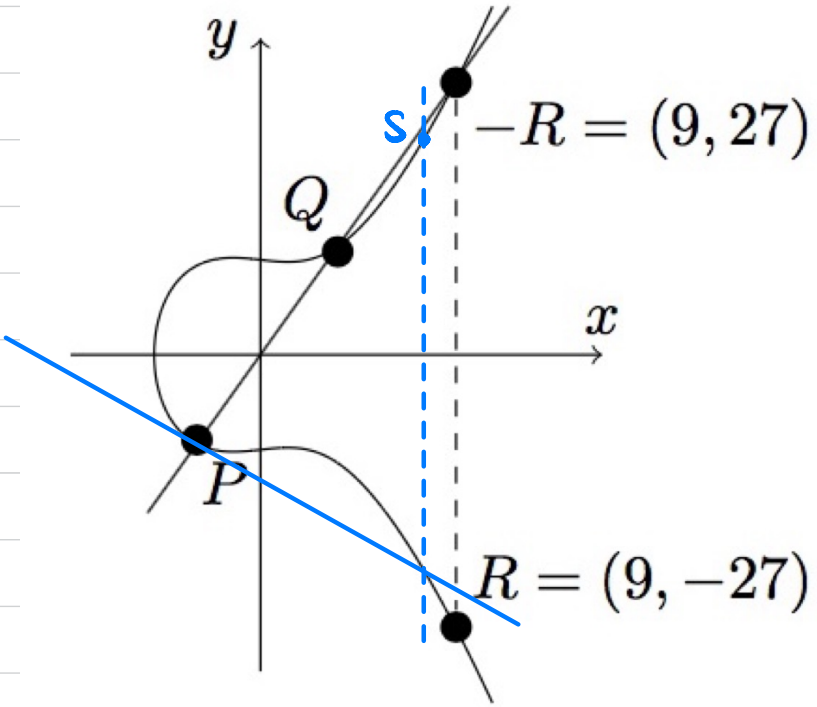
How to find rational points $(x, y) \in \mathbb{Q}^2$ on the curve?

$$x = \frac{s}{t}, \quad y = \frac{u}{v}$$

$s, t, u, v \in \mathbb{Z}$

Elliptic Curves

How to find rational points $(x, y) \in \mathbb{Q}^2$ on the curve?



Example: $y^2 = x^3 - x + 9$

① Chord method $\quad R := P \boxplus Q$

$P = (-1, -3)$
$Q = (1, 3)$
$\Rightarrow \quad y = 3x$

$\Downarrow$

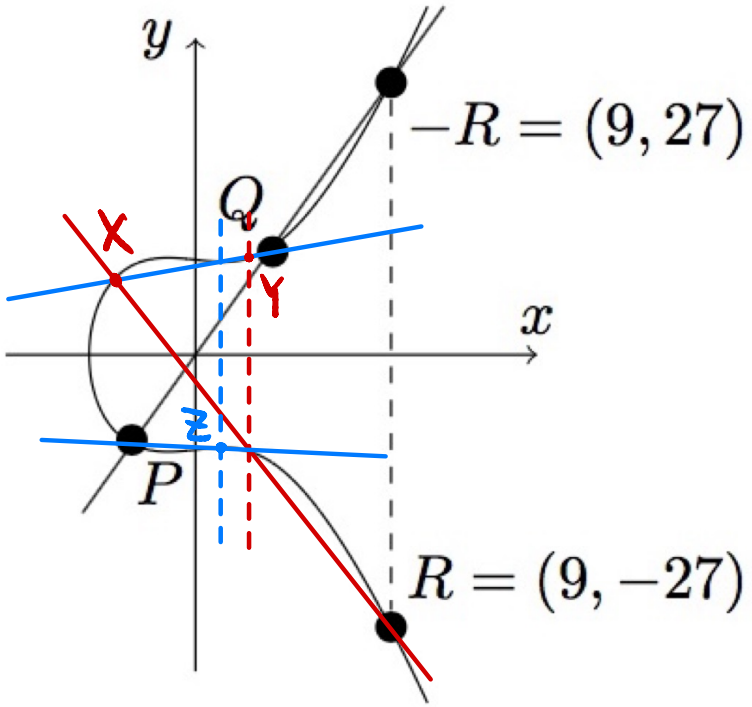
$(3x)^2 = x^3 - x + 9$

$x^3 - 9x^2 - x + 9 = 0$

Why is the third root rational?

② tangent method $\quad S := P \boxplus P$

Elliptic Curves



$R := P \boxplus Q$

$(P \boxplus Q) \boxplus X = P \boxplus (Q \boxplus X)$

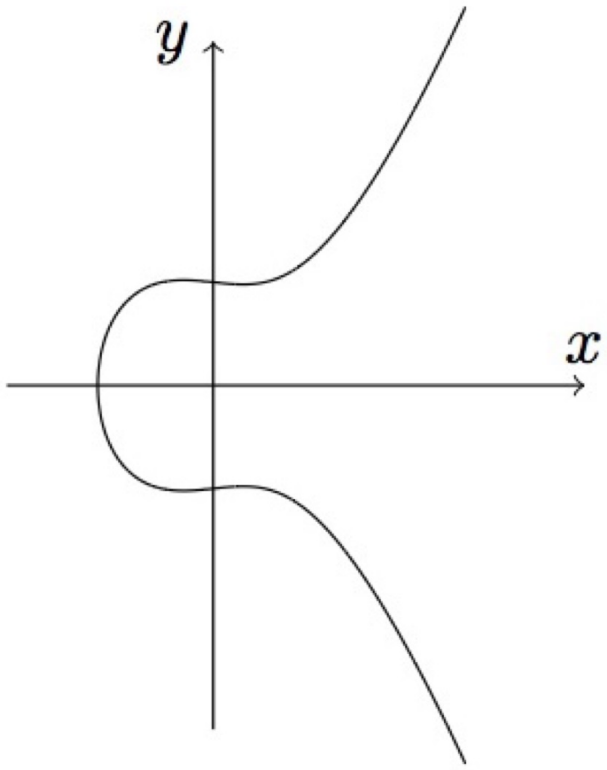
$R = P \boxplus Q$ $\quad Z = Q \boxplus X$

$Y = R \boxplus X$ $\quad Y = P \boxplus Z$

$P \boxplus Q = Q \boxplus P$

Example: $y^2 = x^3 - x + 9$

Elliptic Curves over Finite Fields



$y^2 = x^3 + ax + b$

$(4a^3 + 27b^2 \neq 0)$

Finite field $\mathbb{F}_p$, $p > 3$ prime ($\{0, 1, \cdots, p-1\}$, $+$, $\cdot$, inverse)

Elliptic cure E defined over $\mathbb{F}_p$: $E/\mathbb{F}_p$.

$a, b \in \mathbb{F}_p$

(x, y) is a point on the cure if

$x, y \in \mathbb{F}_p$

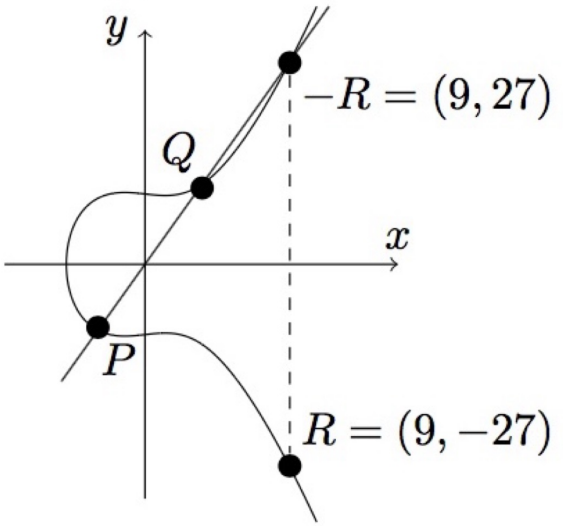
$y^2 = x^3 + ax + b$ over $\mathbb{F}_p$

Point at infinity: O

Example: $y^2 = x^3 + 1$ over $\mathbb{F}_{11}$.

$E/\mathbb{F}_{11} = \{O, (-1, 0), (0, \pm 1), (2, \pm 3), (5, \pm 4), (7, \pm 5), (9, \pm 2)\}$

Elliptic Curves over Finite Fields



Group properties:

① Closure: $\forall g, h \in G,\ g \circ h \in G$ (the $\circ$ is the $+$ operation)

② Existence of an identity:

$\exists e \in G$ s.t. $\forall g \in G,\ e \circ g = g \circ e = g$.

③ Existence of inverse:

$\forall g \in G,\ \exists h \in G$ s.t. $g \circ h = h \circ g = e$

④ Associativity:

$\forall g_1, g_2, g_3 \in G,\ (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$

⑤ Commutativity (abelian):

$\forall g, h \in G,\ g \circ h = h \circ g$

SEA algorithm: count number of points on $E/\mathbb{F}_p$ in time $\text{polylog}(p)$.

How to compute g^a for $a \in \mathbb{Z}_q$?

Elliptic Curve Cryptography

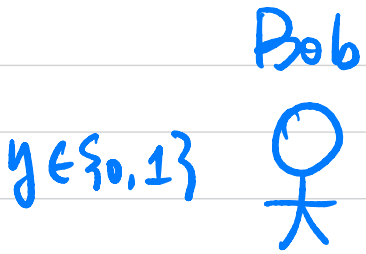
- Curve secp256r1 (P256)
  - prime $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
  - $y^2 = x^3 - 3x + b$ b: 255-bit
  - Number of points on the curve is prime (close to p)
  - Generator point G

- Curve secp256k1
- Curve 25519

Secure Multi-Party Computation

Second date?

$f(x,y) = x \wedge y$



Who is richer?

x

$$f(x,y) = \begin{cases} 0 & \text{if } x < y \\ 1 & \text{Otherwise} \end{cases}$$

y

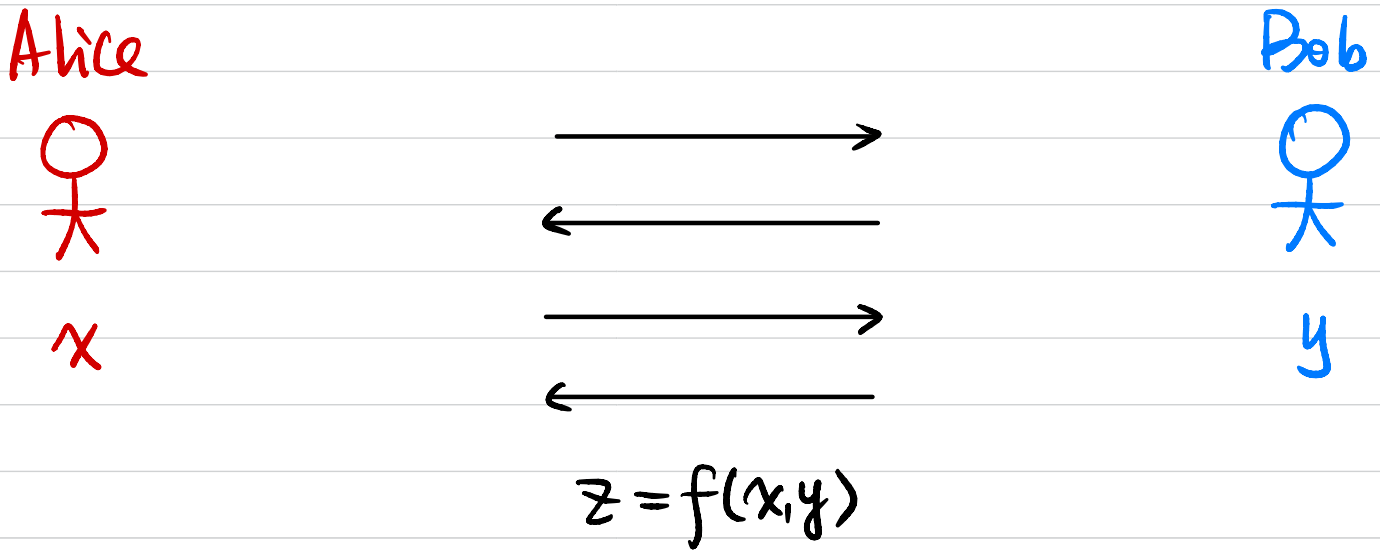
Mutual friends?

X

$f(X,Y) = X \cap Y$

Y

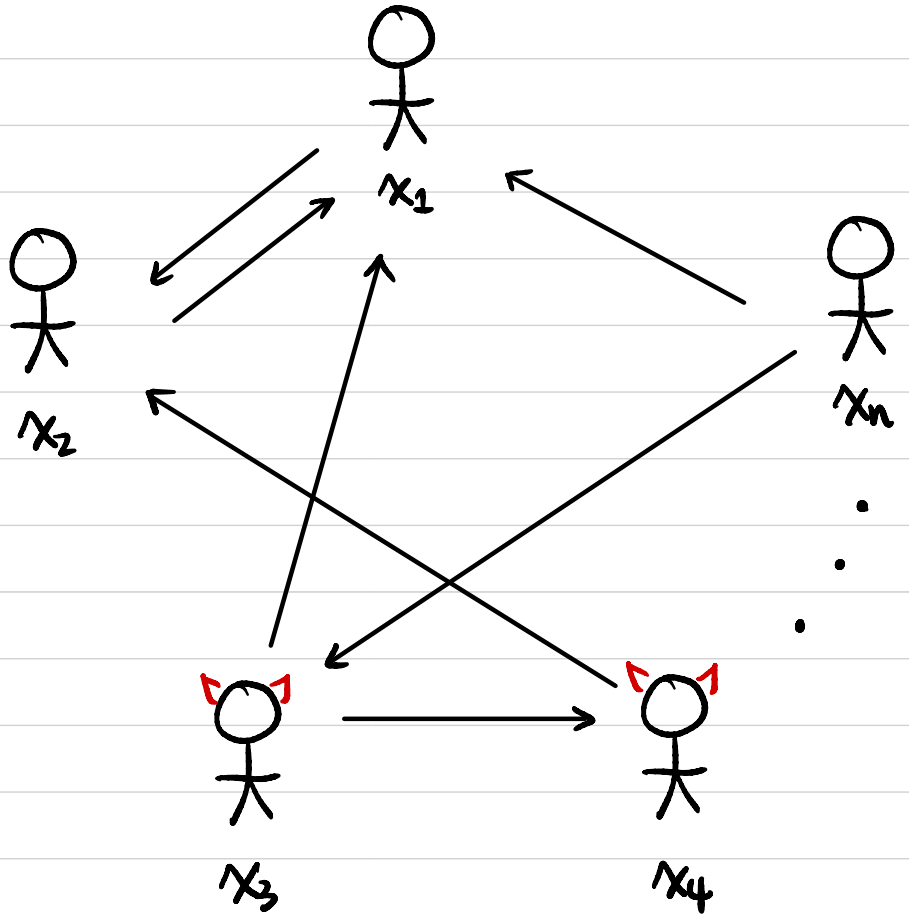
Secure Two-Party Computation (2PC)



Applications:

- Password Breach Alert (Chrome / Firefox / Azure / iOS Keychain)
- Privacy-Preserving Contact Tracing for COVID-19 (Apple & Google)
- Ads Conversion Measurements / Personalized Advertising (Google / Meta)

Secure Multi-Party Computation (MPC)



$$z = f(x_1, \cdots, x_n)$$

Secure Multi-Party Computation (MPC)

Applications:

- Privacy-Preserving Inventory Matching (J.P. Morgan)
- Setup Ceremony to securely generate CRS (Zcash)
- Distributed Key Management (Unbound / Coinbase)
- Federated Learning (Google Keyboard Search Suggestion)
- Auctions (Danish sugar beet auction)
- Boston gender wage gap (Boston Women's Workforce Council)

- Study / Analysis on Medical Data
- Fraud / Money Laundering Detection (banks)

Setting

- n parties $P_1, P_2, \cdots, P_n$
  with private inputs $x_1, x_2, \cdots, x_n$

- Jointly compute $f(x_1, x_2, \cdots, x_n)$

- Communication:
  Authenticated secure point-to-point channels between each pair (P_i, P_j)
  (sometimes also assume broadcast channel)

- The adversary can "corrupt" a subset of the parties
  (e.g. at most t parties)

What properties do we want?