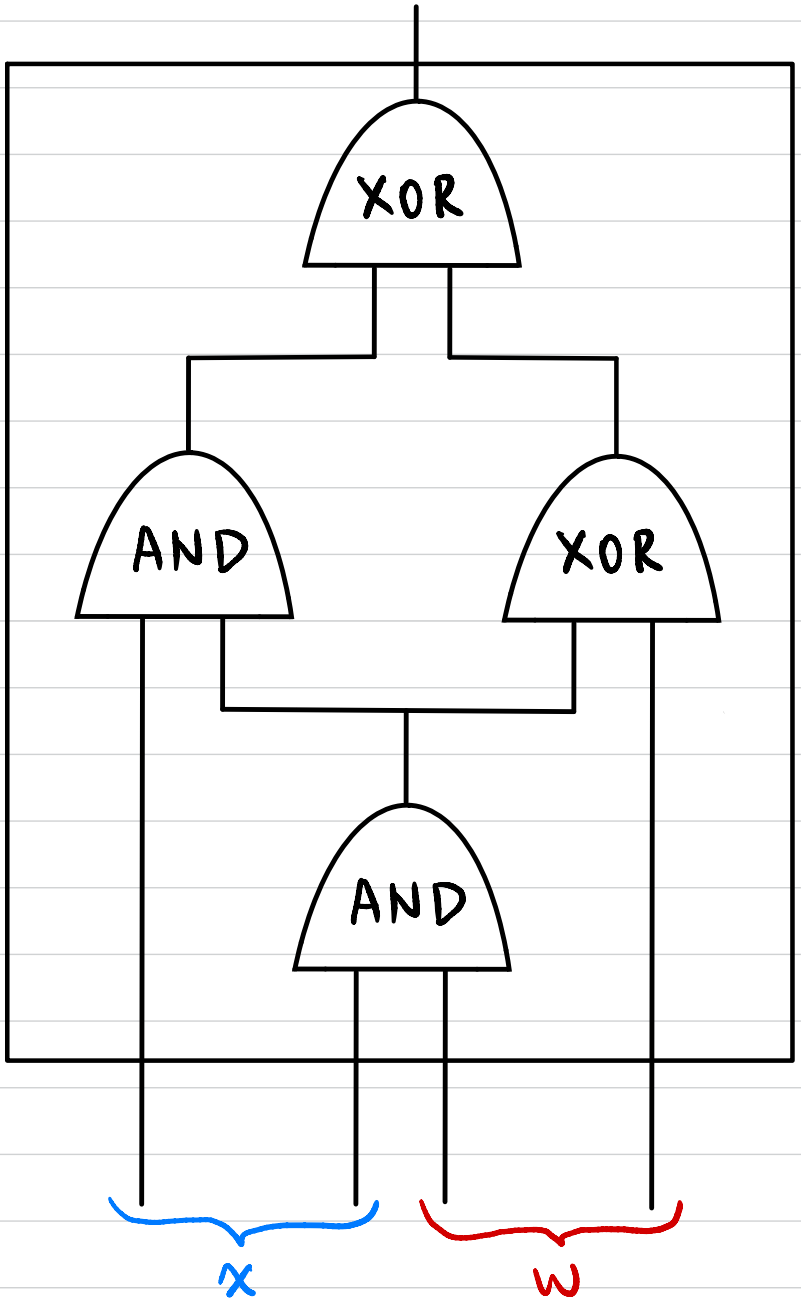
CSCI 1515 Applied Cryptography

This Lecture:

- Succinct Non-Interactive Arguments (SNARGs)
- SNARGs from PCP
- Blockchain & Cryptocurrencies

Circuit Satisfiability (NP Complete)



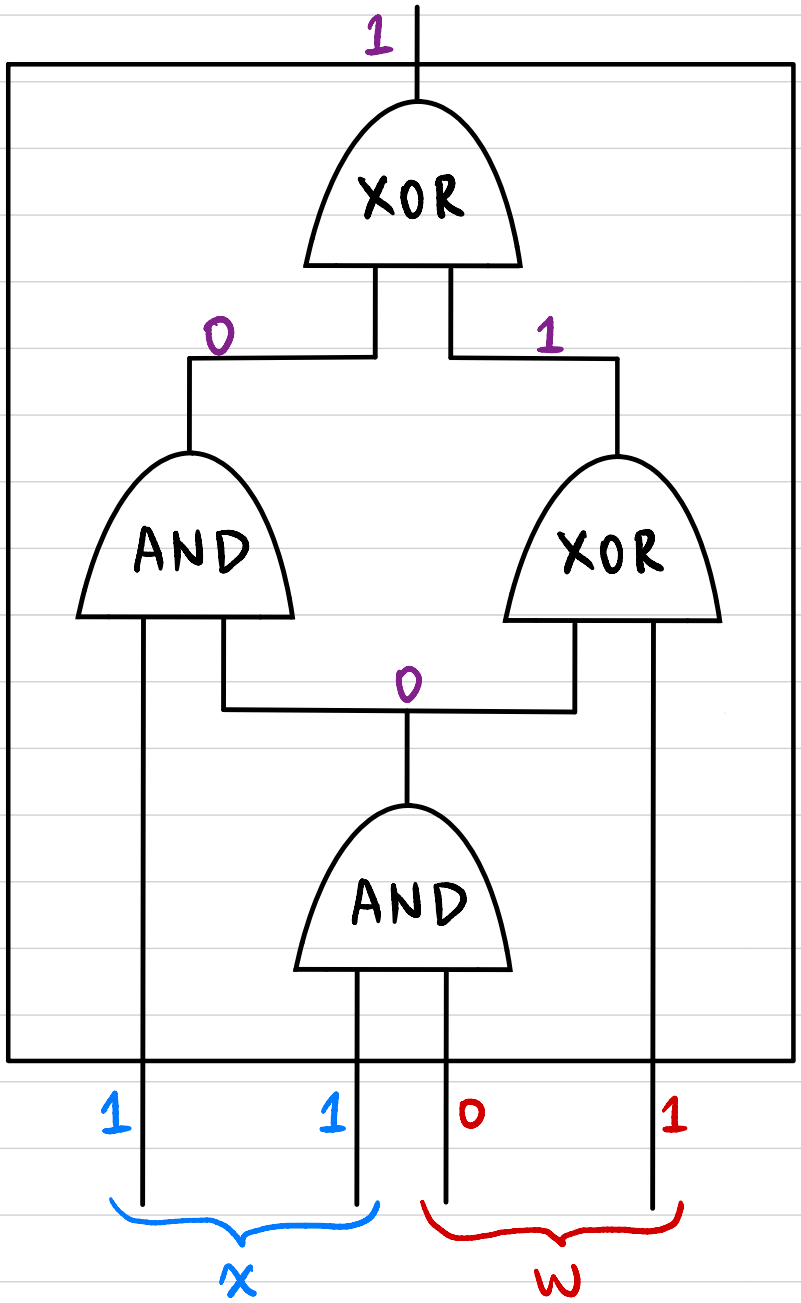
NP language $L_C = \{x \in \{0,1\}^n : \exists w \in \{0,1\}^m \text{ s.t. } C(x,w) = 1\}$

NP relation $R_{L_C} = \{(x, w) : C(x, w) = 1\}$

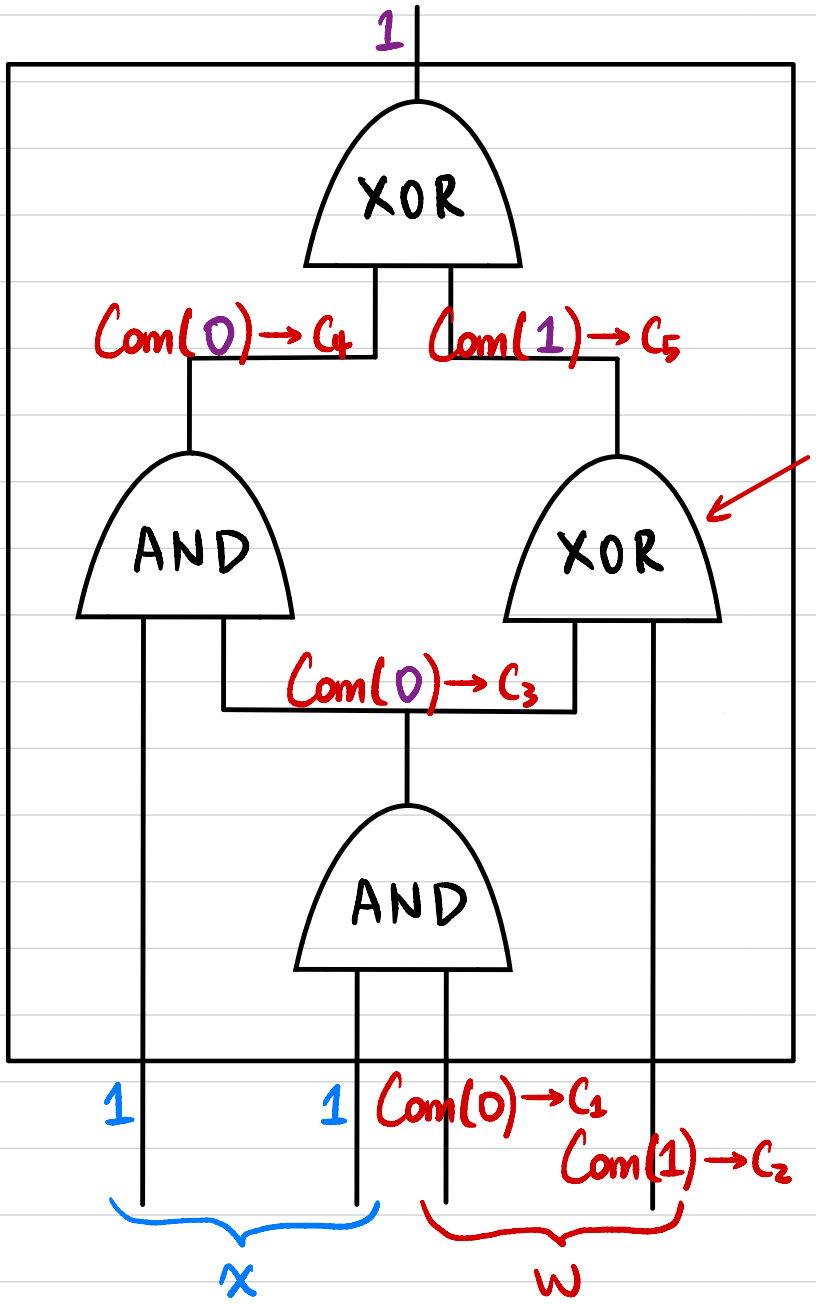
(public) statement

(secret) witness

ZKP for Circuit Satisfiability



ZKP for Circuit Satisfiability



$$\begin{pmatrix} C_2 = Com(0) \\ C_3 = Com(0) \\ C_5 = Com(0) \end{pmatrix}$$

OR

$$\begin{pmatrix} C_2 = Com(1) \\ C_3 = Com(0) \\ C_5 = Com(1) \end{pmatrix}$$

OR

$$\begin{pmatrix} C_2 = Com(1) \\ C_3 = Com(1) \\ C_5 = Com(0) \end{pmatrix}$$

OR

$$\begin{pmatrix} C_2 = Com(0) \\ C_3 = Com(1) \\ C_5 = Com(1) \end{pmatrix}$$

Proof Systems for Circuit Satisfiability

NP relation $R_{\mathcal{C}} = \{(x, w) : C(x, w) = 1\}$

	NP	Σ-Protocol	(Fiat-Shamir) NIZK
	$P(x, w) \xrightarrow{w} V(x)$	$P(x, w) \rightleftarrows V(x)$	$P(x, w) \xrightarrow{\pi} V(x)$
Zero-Knowledge	NO	YES	YES
Non-Interactive	YES	NO	YES
Communication	$O(\|w\|)$	$O(\|C\| \cdot \lambda)$	$O(\|C\| \cdot \lambda)$
V's computation	$O(\|C\|)$	$O(\|C\|)$	$O(\|C\|)$

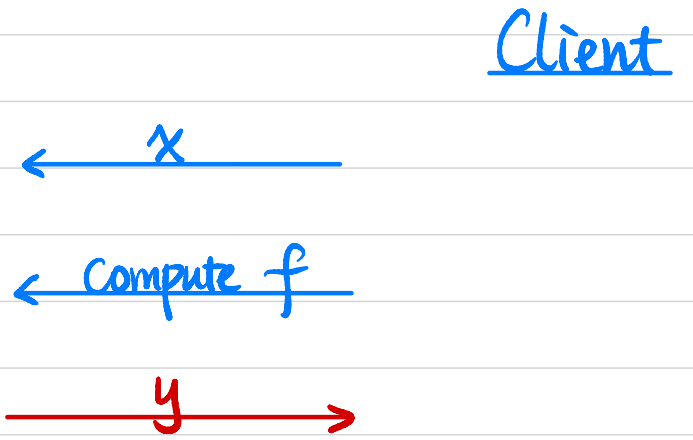
Can we have communication complexity & verifier's computational complexity sublinear in $|C|$ & $|w|$?

Succinct Non-Interactive Argument

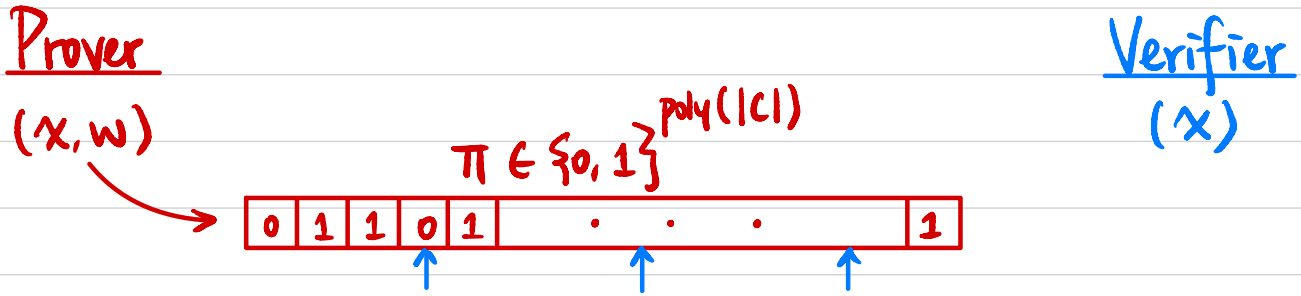
- **SNARG:** Succinct Non-Interactive Argument
- **SNARK:** Succinct Non-Interactive Argument of Knowledge
- **zk-SNARG / zk-SNARK:** SNARG / SNARK + Zero-Knowledge

- **Succint:** $|\pi| = \text{poly}(\lambda, \log |C|)$
  Verifier runtime $\text{poly}(\lambda, |x|, \log |C|)$
- **Argument:** In Soundness / Proof of knowledge: $\forall$ PPT P^*

Verifiable Computation



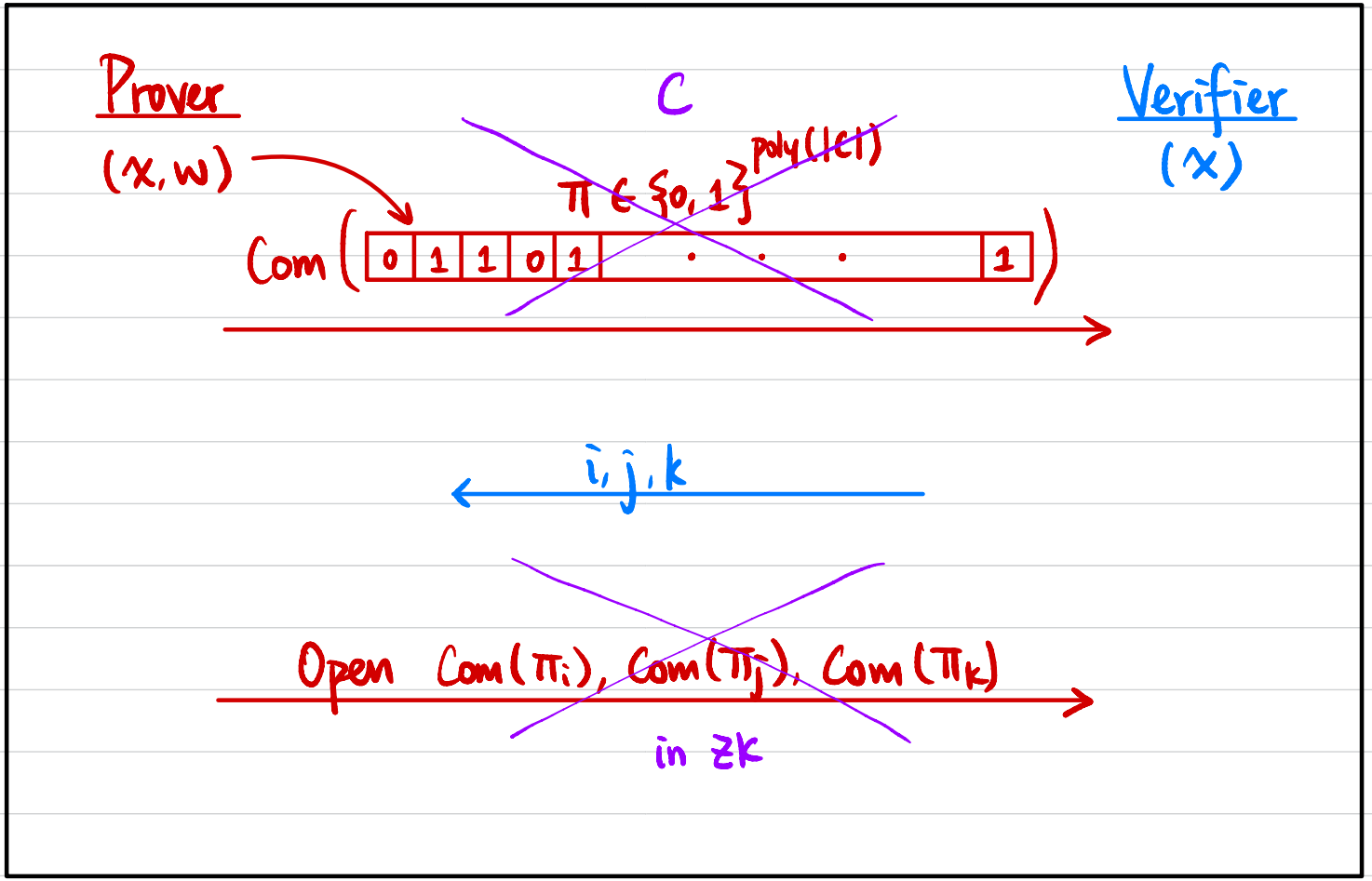
Probabilistically Checkable Proof (PCP)



PCP Theorem (Informal):

Every NP language has a PCP where the verifier reads only a **constant** number of bits of the proof.

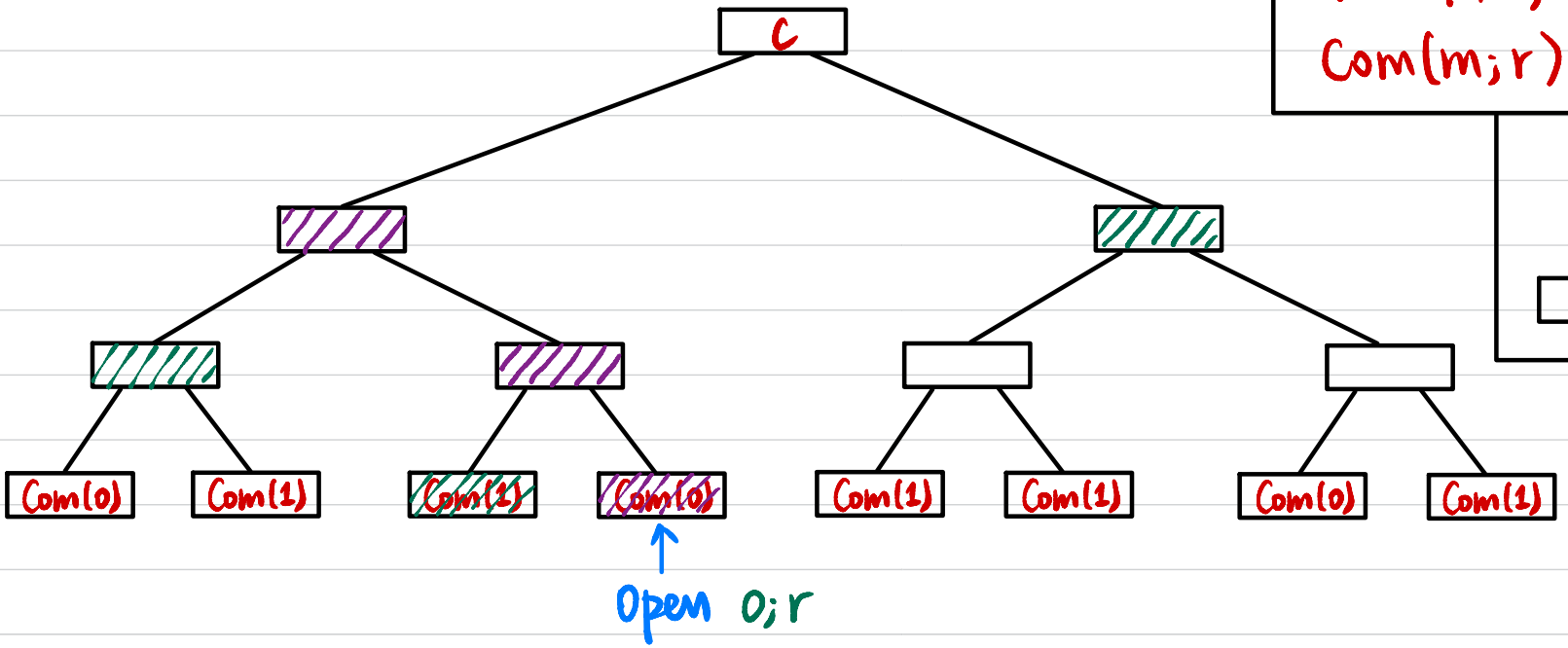
First Attempt



Is it succint? Proof size / communication $\text{poly}(\lambda, \log|C|)$?
Verifier runtime $\text{poly}(\lambda, |x|, \log|C|)$?

Is it zk?

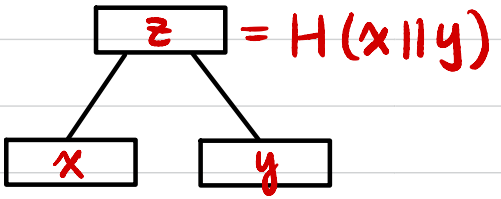
Merkle Tree



Hash-based Commitment

$r \xleftarrow{\$} \{0,1\}^\lambda$

$Com(m; r) := H(r \| m) \rightarrow c$



How to open commitment?

Why hiding & binding?

Hiding of Com

Collision-Resistance of H

Transactions in Real Life

Alice → Starbucks $3

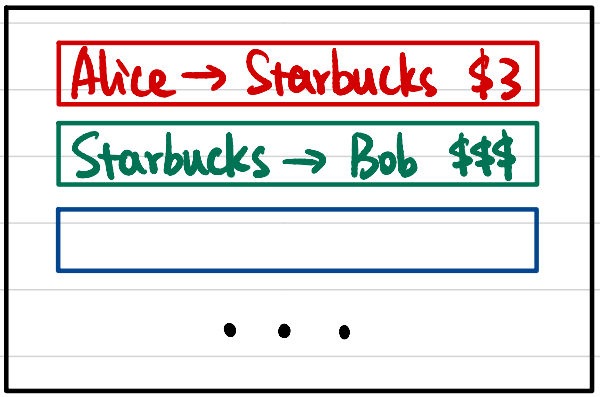
Starbucks → Bob $$$

Bank of America

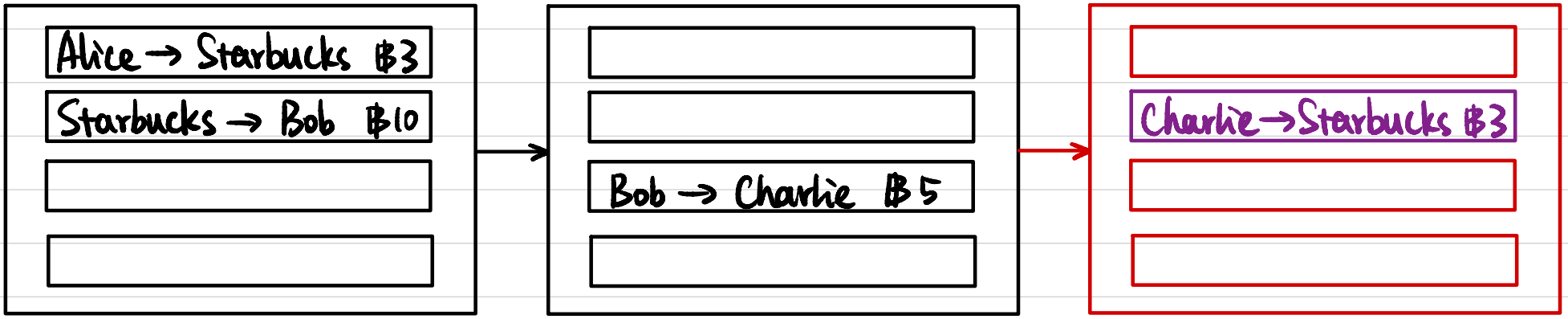
① initiated by sender

② enough balance in sender's account

A trusted party that maintains a private ledger



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 ← How?
  - ② enough balance in sender's account
- Agree on next block ← How?

Step 3: Repeat

Transaction Authentication

Alice: $(vk_A, sk_A) \leftarrow KeyGen(1^\lambda)$

Bob: $(vk_B, sk_B) \leftarrow KeyGen(1^\lambda)$

Charlie: $(vk_C, sk_C) \leftarrow KeyGen(1^\lambda)$

Starbucks: $(vk_S, sk_S) \leftarrow KeyGen(1^\lambda)$

Bob → Charlie ฿5 :

$m_1 = (vk_B, vk_C, 5) \quad \sigma_1 \leftarrow Sign_{sk_B}(m_1)$

Charlie → Starbucks ฿3 :

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

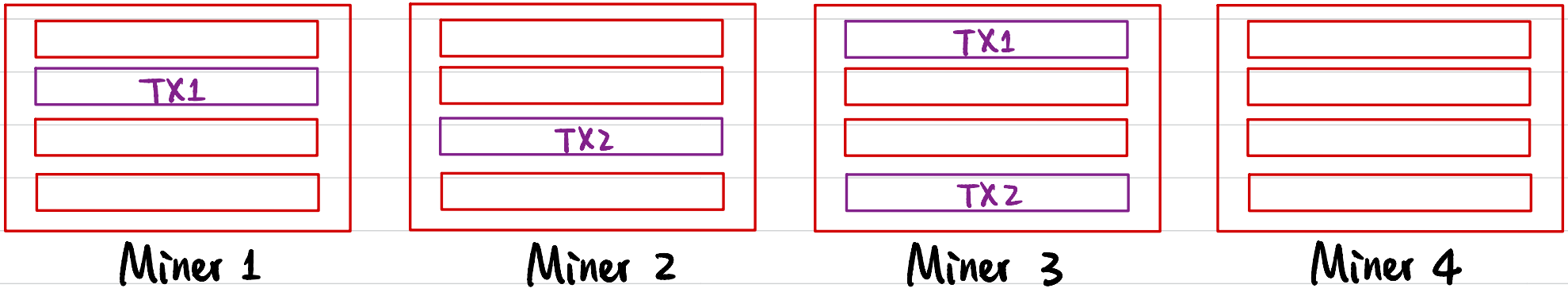
Consensus Protocol

TX1 = Charlie → Starbucks ฿3 :

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

TX2 = Charlie → Alice ฿4 :

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

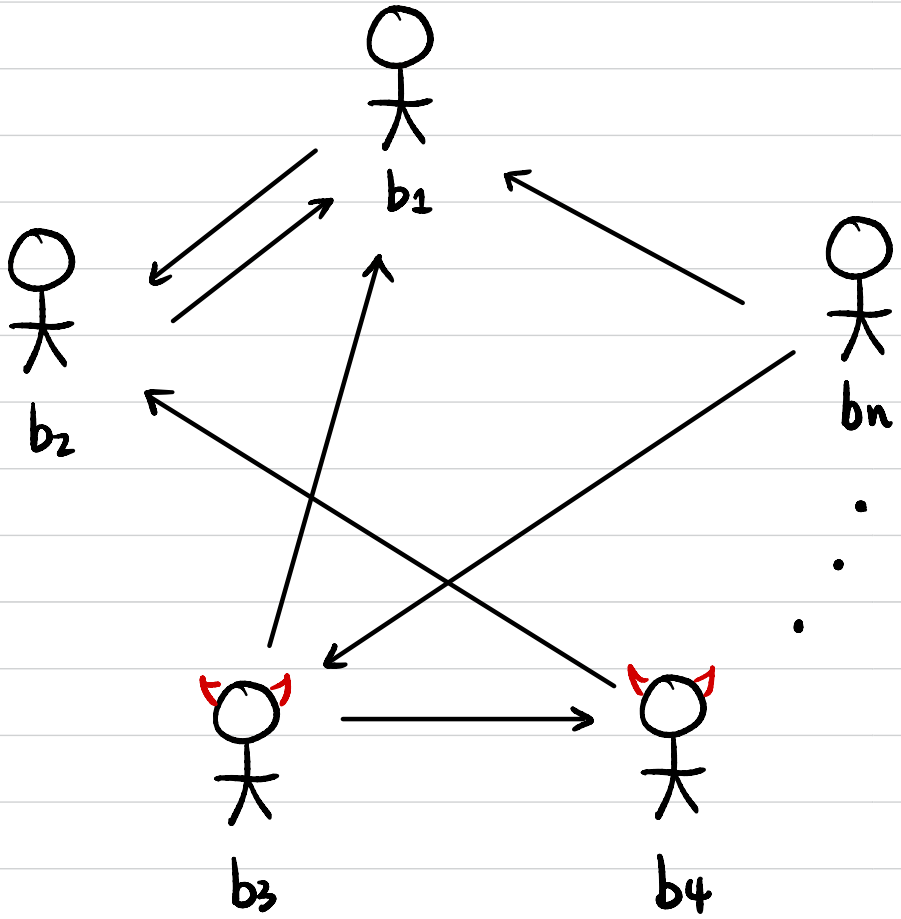


"permissionless"

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:

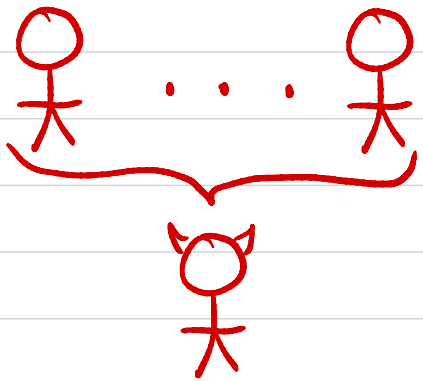
If $n \geq 3t+1$, (necessary)

then it's possible to reach consensus.

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

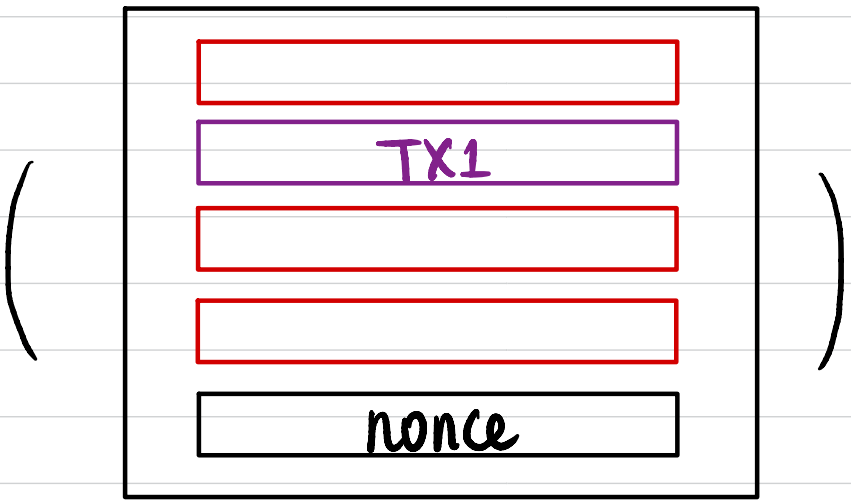
Any problem?

"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.