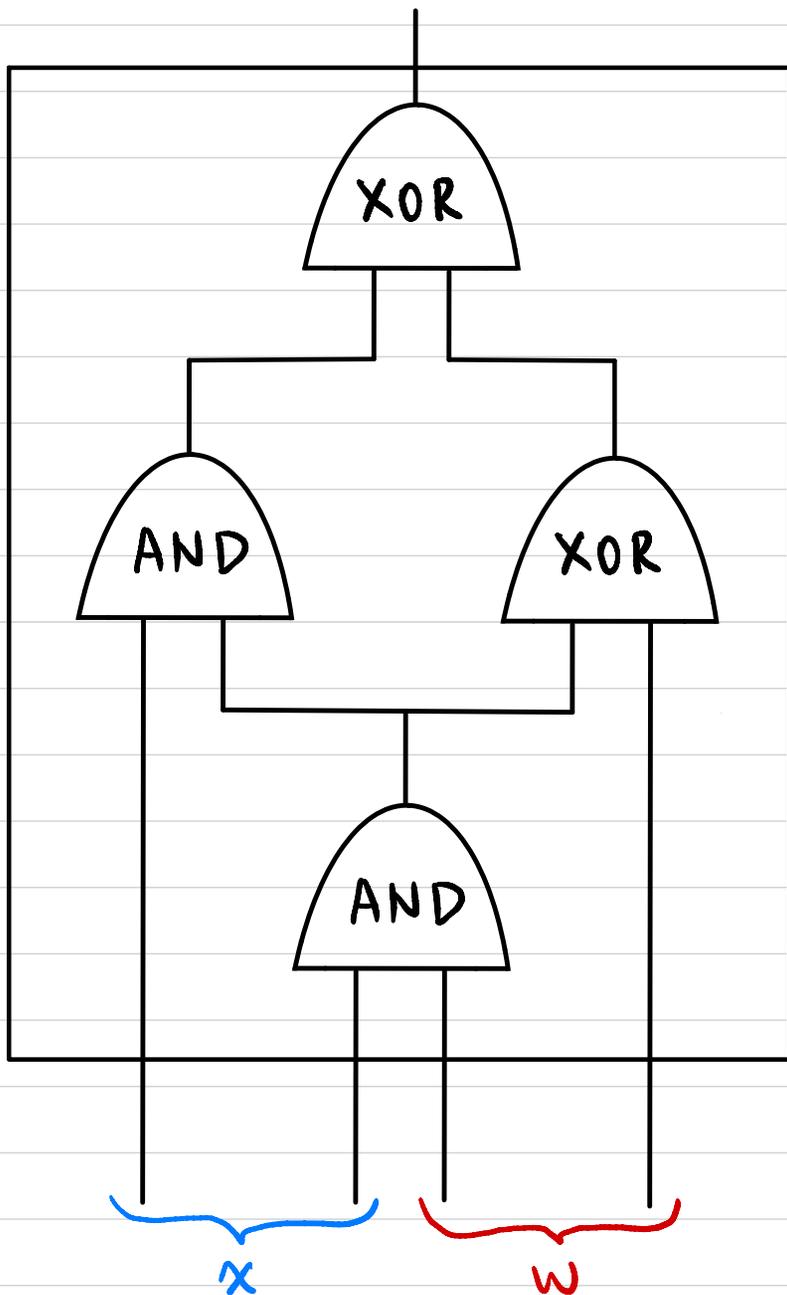


CSCI 1515 Applied Cryptography

This Lecture:

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

Circuit Satisfiability (NP Complete)

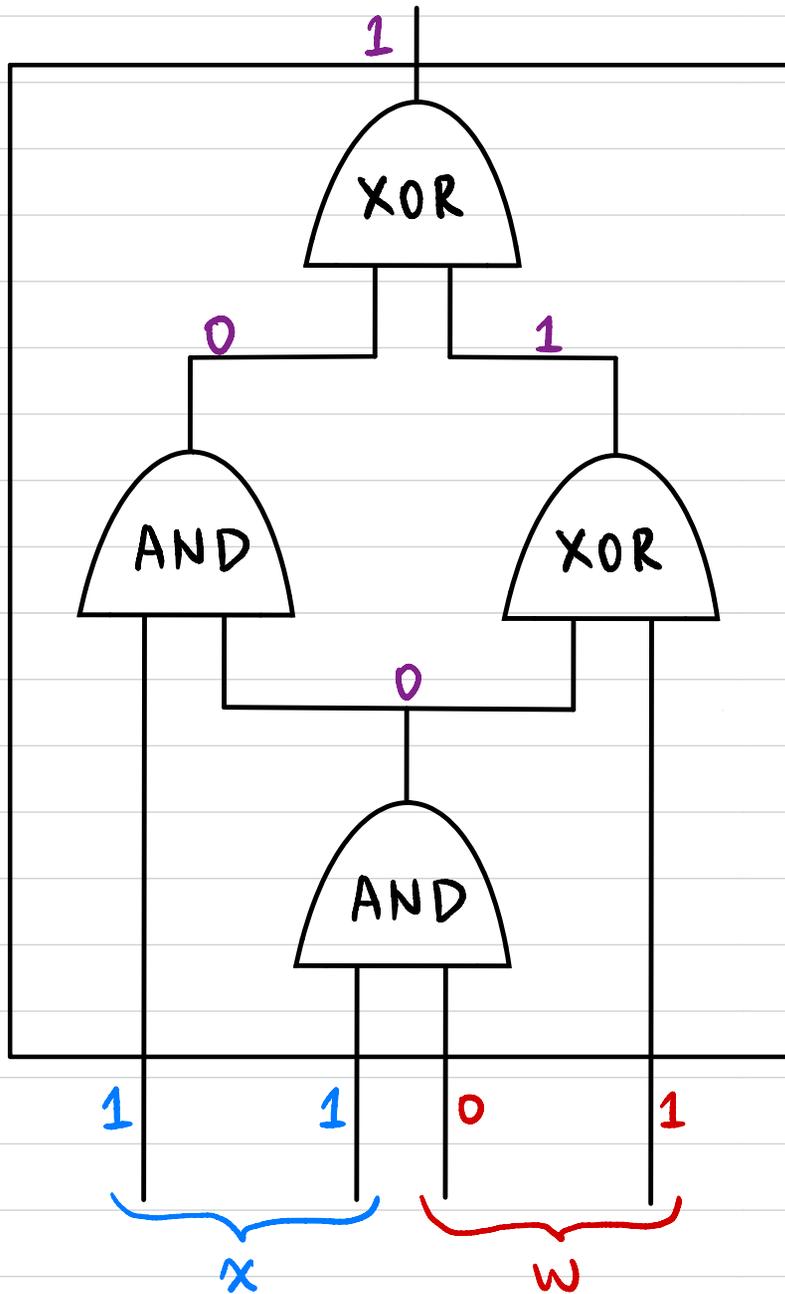


NP language $L_C = \{x \in \{0,1\}^n : \exists w \in \{0,1\}^m \text{ st. } C(x,w) = 1\}$

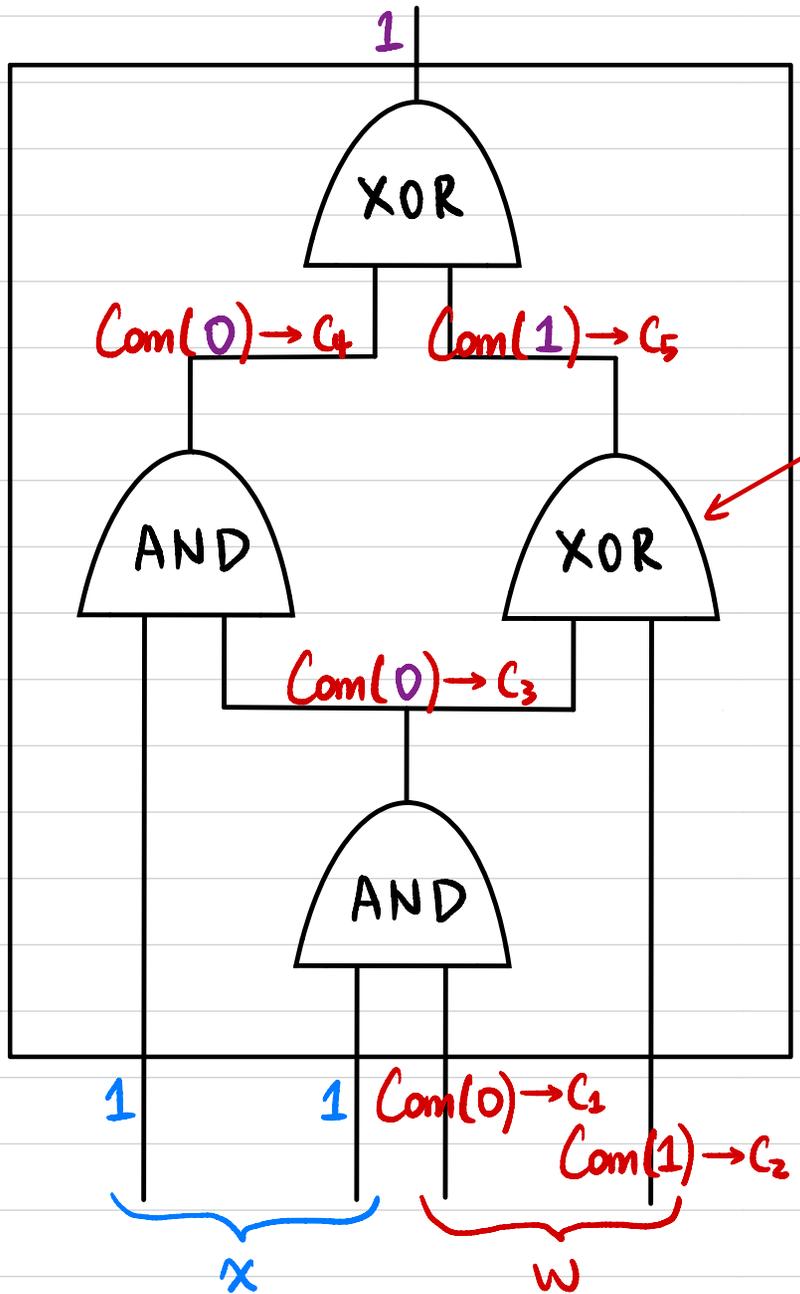
NP relation $R_C = \{(x, w) : C(x, w) = 1\}$

(public) (secret)
Statement 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 $R_C = \{ (x, w) : C(x, w) = 1 \}$

	NP	Σ -Protocol	(Fiat-Shamir) NIZK
	$P(x, w) \xrightarrow{w} V(x)$	$P(x, w) \begin{matrix} \xrightarrow{\quad} \\ \xleftarrow{\quad} \\ \xrightarrow{\quad} \end{matrix} V(x)$	$P(x, w) \xrightarrow{\pi} V(x)$
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
- **Succinct**: $|\pi| = \text{poly}(\lambda, \log |C|)$
Verifier runtime $\text{poly}(\lambda, |x|, \log |C|)$
- **Argument**: In Soundness / Proof of Knowledge: $\forall \text{PPT } P^*$

Verifiable Computation

Server

Client

← x

← Compute f

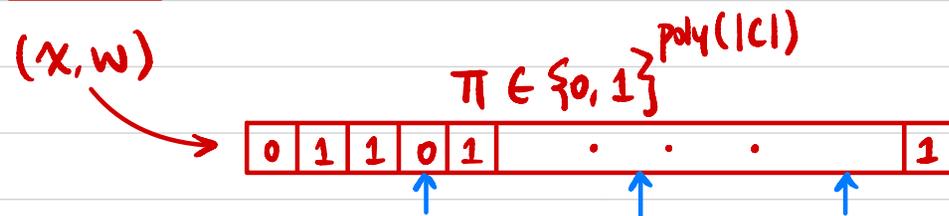
→ y

$$y \stackrel{?}{=} f(x)$$

Probabilistically Checkable Proof (PCP)

Prover

(x, w)



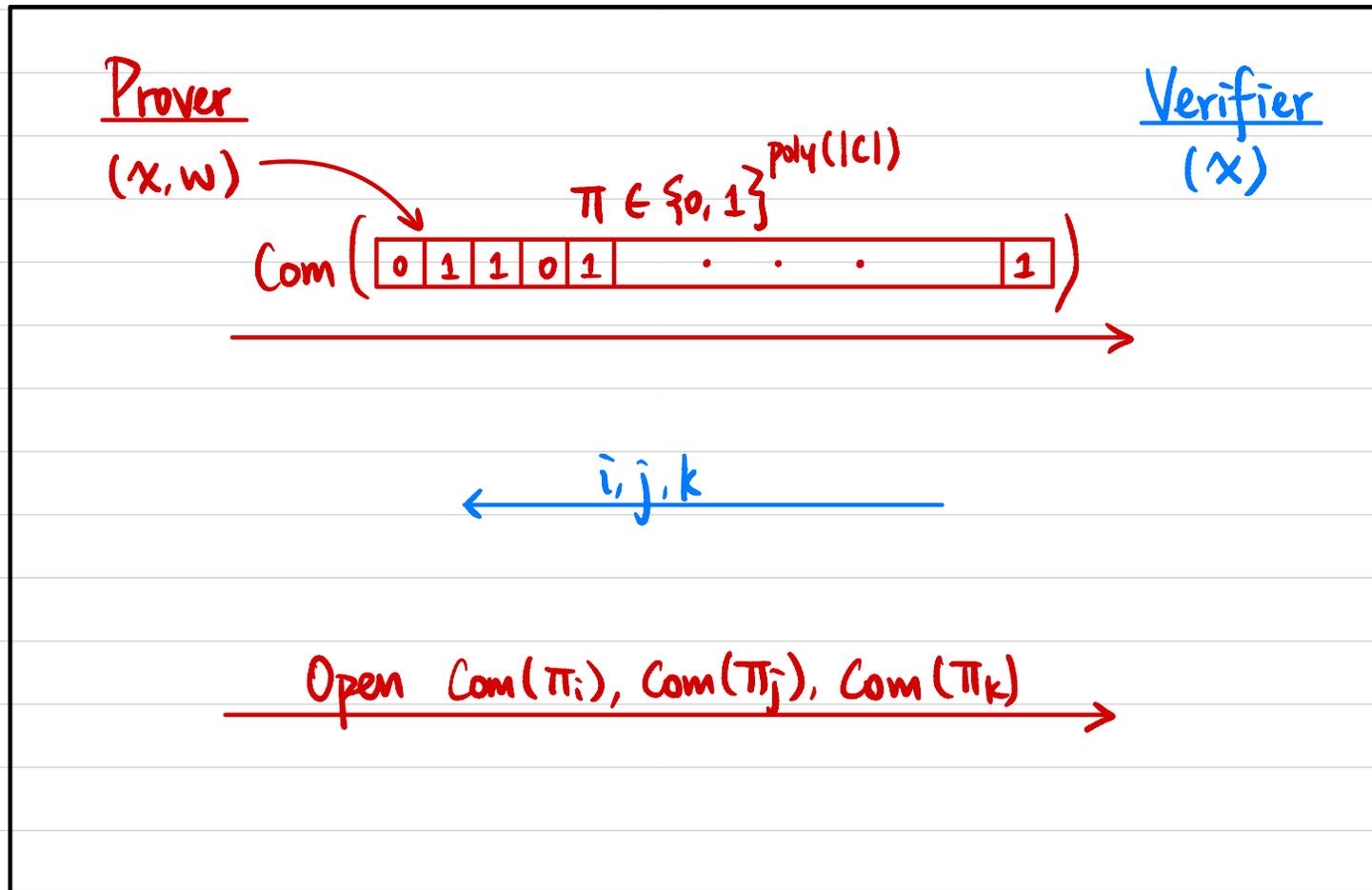
Verifier

(x)

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 succinct?

Proof size / communication $\text{poly}(\lambda, \log |C|)$?

Verifier runtime $\text{poly}(\lambda, |x|, \log |C|)$?

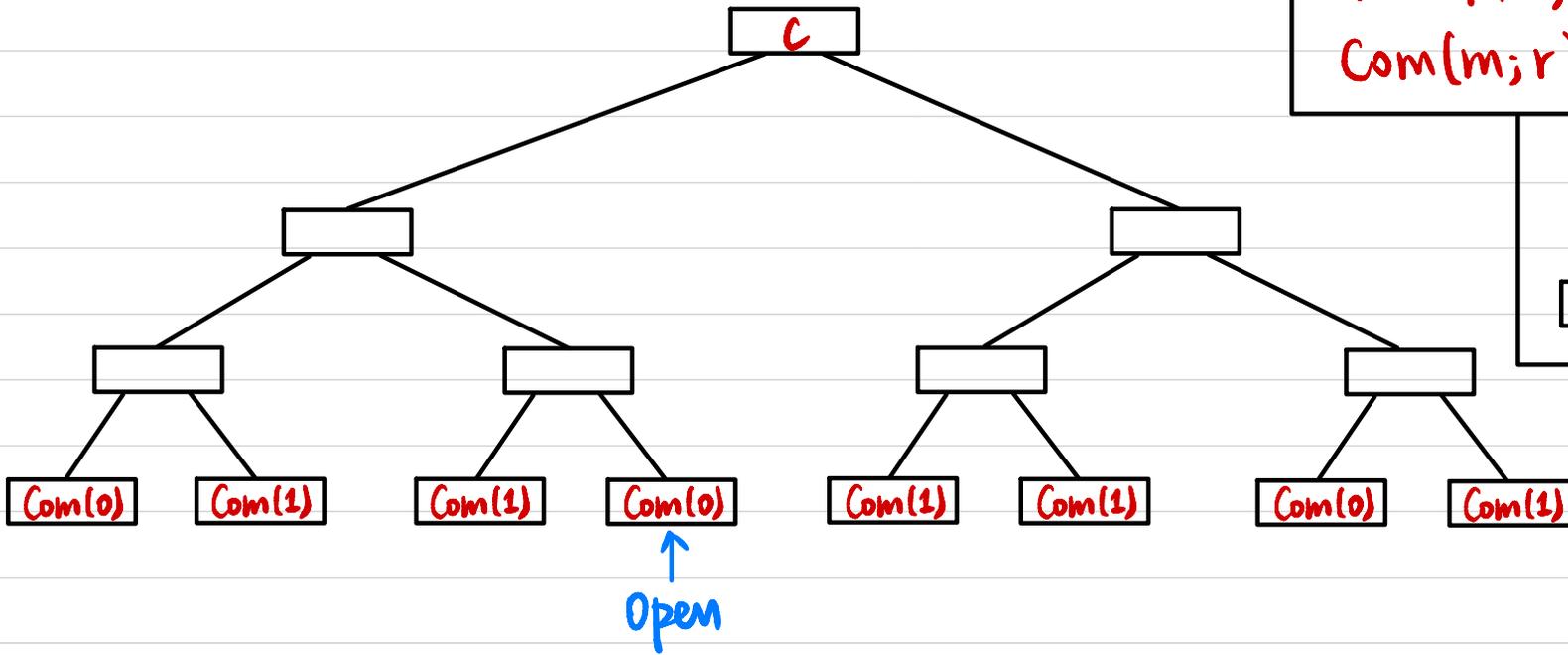
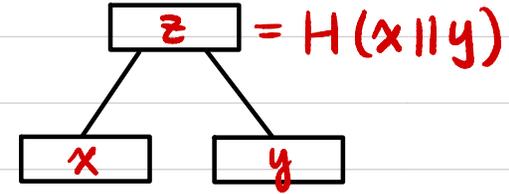
Is it zk?

Merkle Tree

Hash-based Commitment

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

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



How to open commitment?

Why hiding & binding?

Transactions in Real Life

Alice



Alice → Starbucks \$3

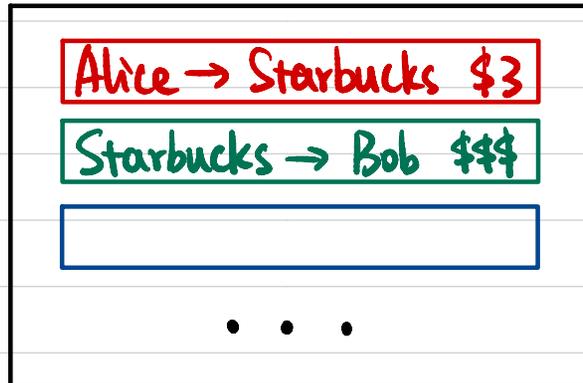
Starbucks → Bob \$\$\$

Bank of America

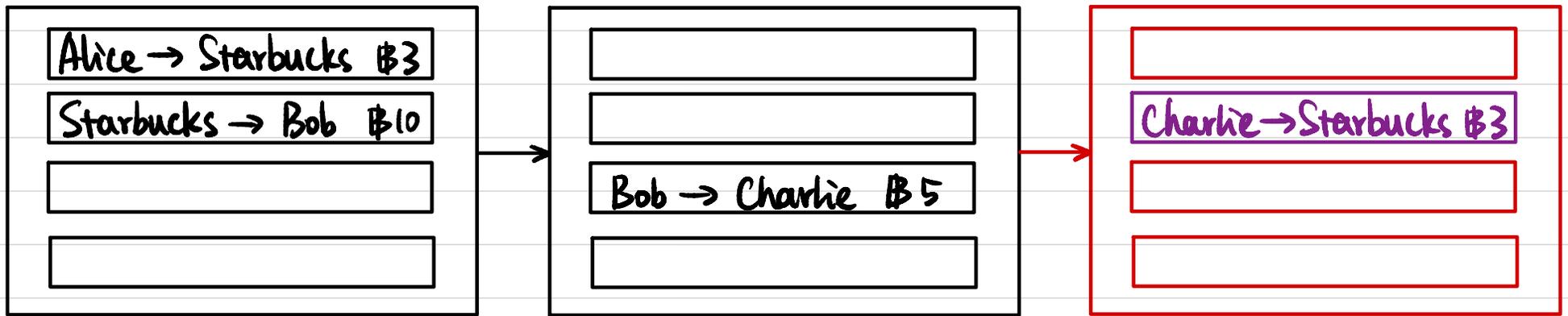


- ① initiated by sender
- ② enough balance in sender's account

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: $(VK_A, SK_A) \leftarrow \text{KeyGen}(1^\lambda)$

Bob: $(VK_B, SK_B) \leftarrow \text{KeyGen}(1^\lambda)$

Charlie: $(VK_C, SK_C) \leftarrow \text{KeyGen}(1^\lambda)$

Starbucks: $(VK_S, SK_S) \leftarrow \text{KeyGen}(1^\lambda)$

Bob \rightarrow Charlie \$5 :

$$m_1 = (VK_B, VK_C, 5) \quad \sigma_1 \leftarrow \text{Sign}_{SK_B}(m_1)$$

Charlie \rightarrow Starbucks \$3 :

$$m_2 = (VK_C, VK_S, 3) \quad \sigma_2 \leftarrow \text{Sign}_{SK_C}(m_2)$$

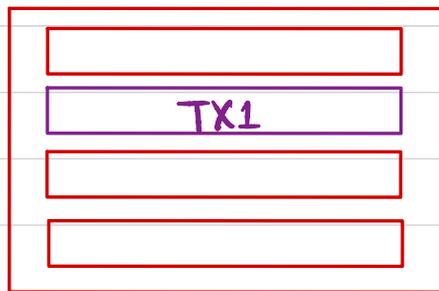
Consensus Protocol

TX1 = Charlie → Starbucks \$3 :

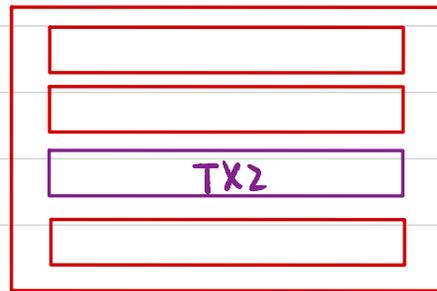
$$m_2 = (vk_c, vk_s, 3) \quad \sigma_2 \leftarrow \text{Sign}_{sk_c}(m_2)$$

TX2 = Charlie → Alice \$4 :

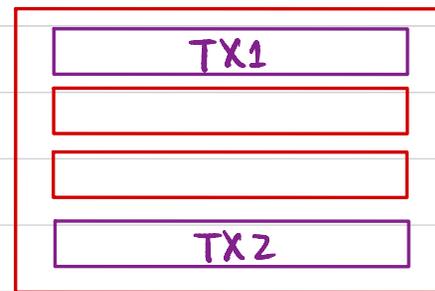
$$m_3 = (vk_c, vk_a, 4) \quad \sigma_3 \leftarrow \text{Sign}_{sk_c}(m_3)$$



Miner 1



Miner 2



Miner 3

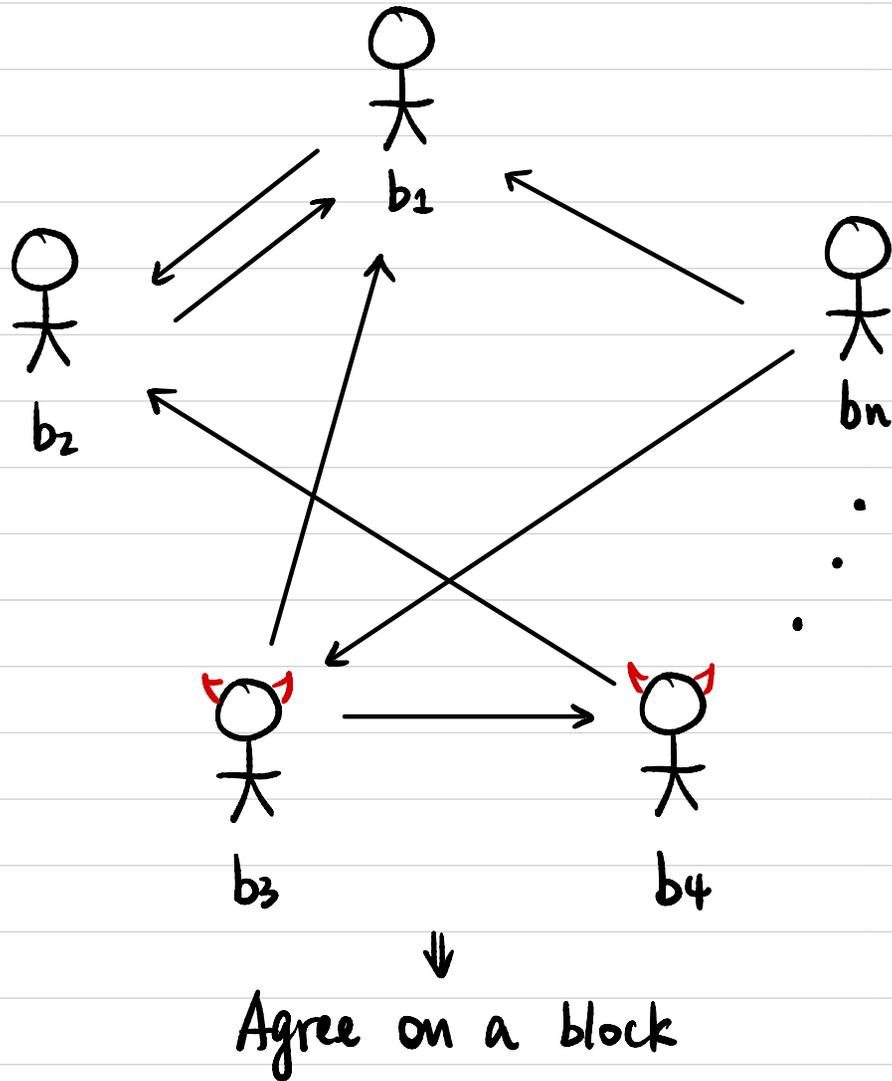


Miner 4

WANT: ① All miners **agree** on the same block

② New block is **valid**

Byzantine Agreement



Byzantine Fault Tolerance (BFT) Protocol:

If $n \geq 3t + 1$,
where n is necessary

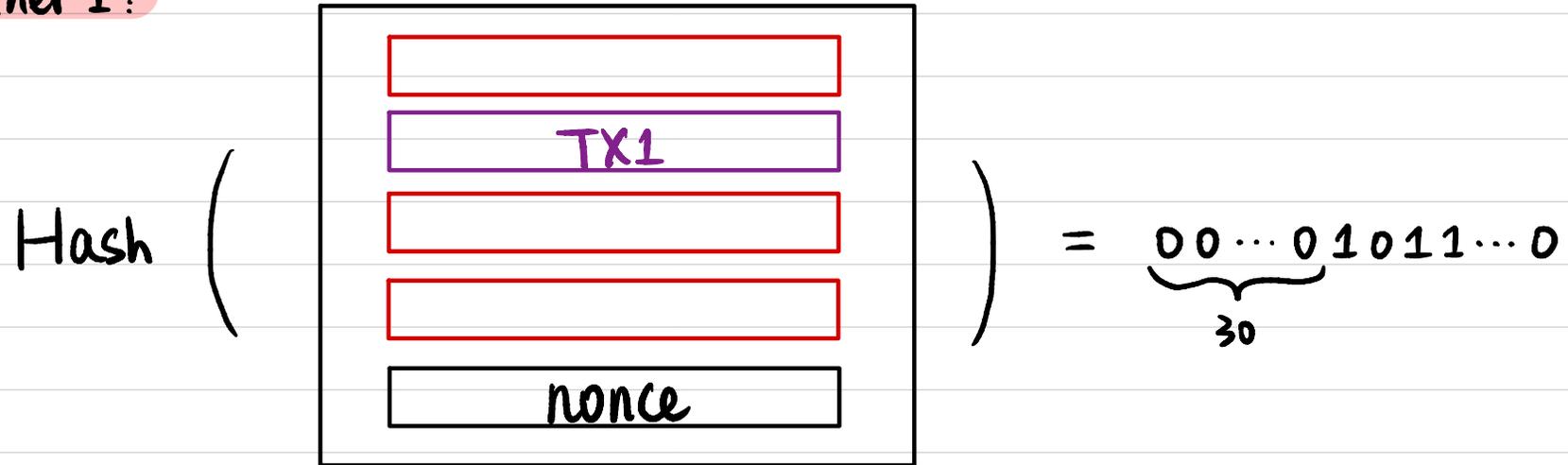
then it's possible to reach consensus.

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

Any problem?

Proof of Work (PoW)

Miner 1:

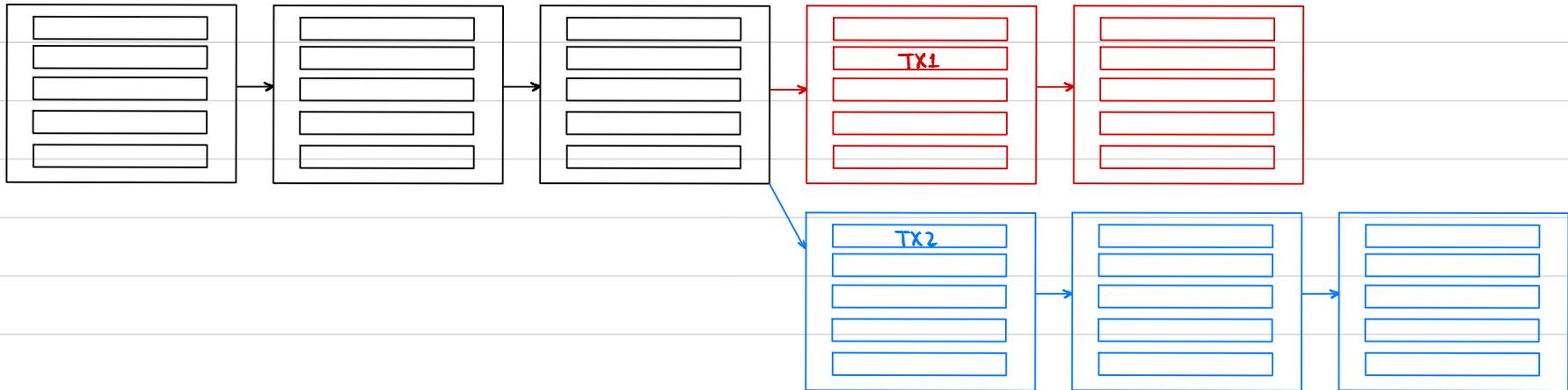
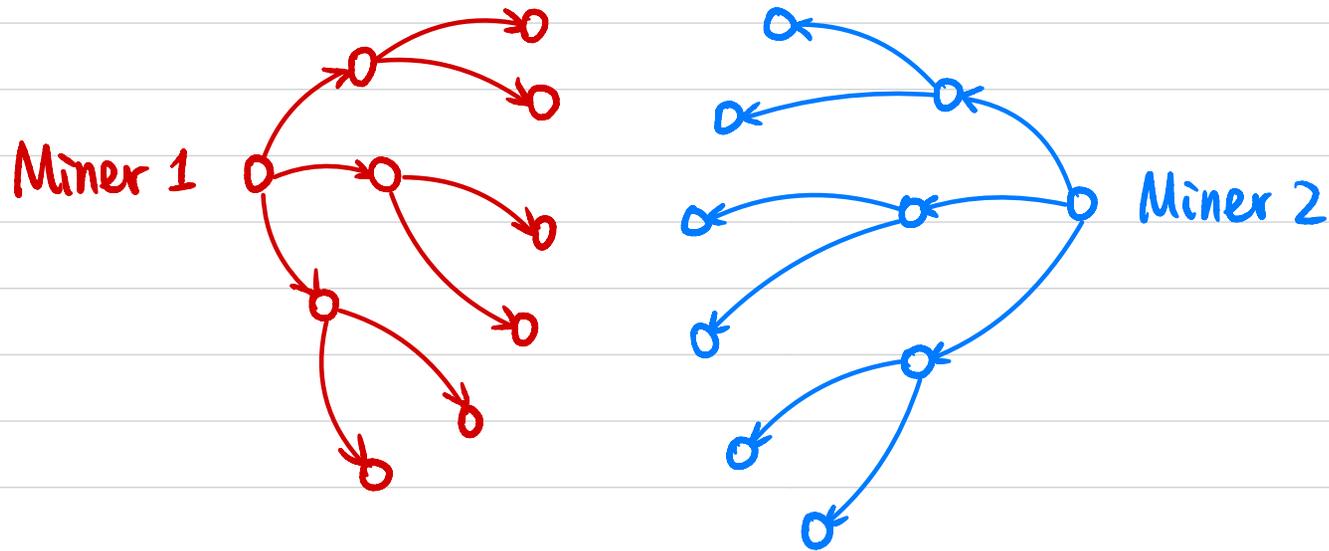


Find nonce s.t. Hash(block) has ≥ 30 leading 0's.

Consensus Protocol:

Whoever first finds a block that hashes to a value w/ ≥ 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 ~ 10 min 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