CS355 Spring 2025

https://cs355.stanford.edu

Applied Zero Knowledge Proofs

Dan Boneh and Binyi Chen

Stanford University

[discussions on edstem, homework on gradescope]

Succinct non-interactive proofs

SNARK: a succinct proof that a certain statement is true

Example statement: "I know an m such that SHA256$(m) = 0$"

- **SNARK**: the proof is **"short"** and **"fast"** to verify

  [if m is 1GB then the trivial proof (the message m) is neither]

- **zk-SNARK**: the proof "reveals nothing" about m

A simple example: digits of Pi

Alice claims that the billion-th digit of Pi is 5

- if Bob, Carol, and David want to check $\Rightarrow$ redo the entire computation

Alternatively: Alice publishes a SNARK proof π for her claim

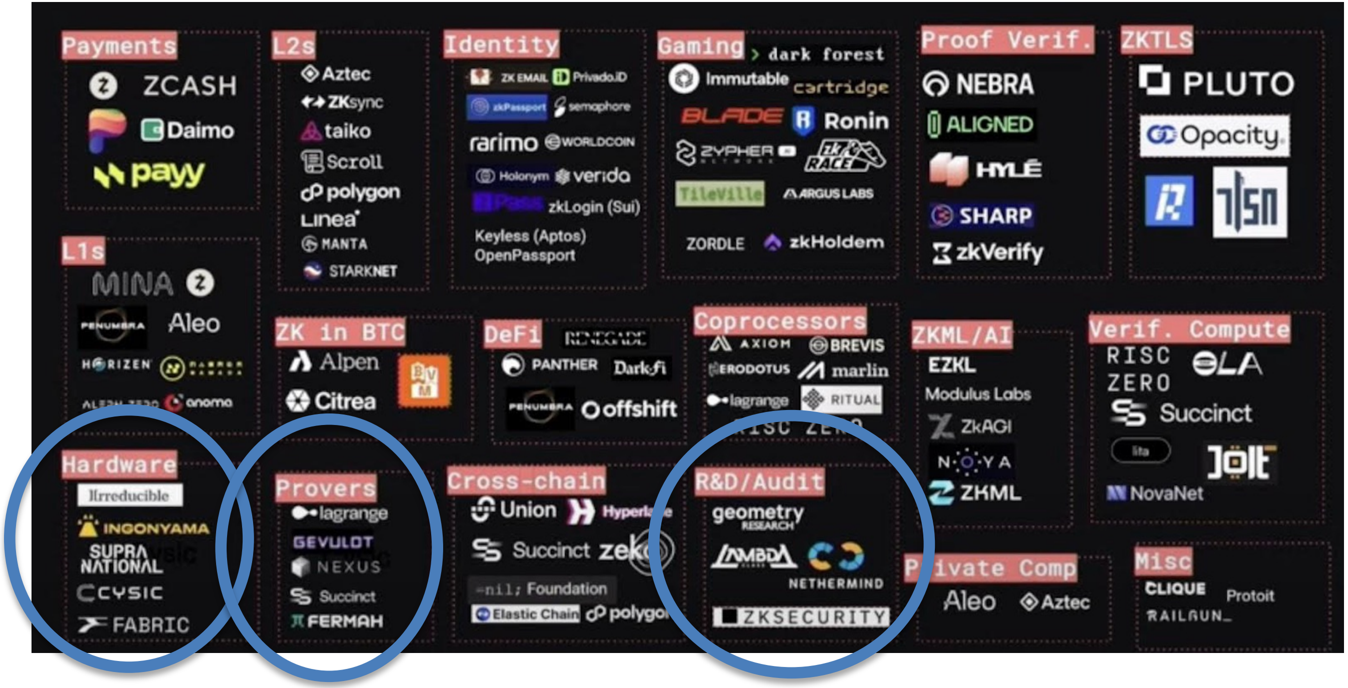
- Now, Bob, Carol, and David can just check the proof π (fast)
- Alice would spend the effort to build π if there are many verifiers

How hard is proof generation? ≈30 MHz RISC-V computer

(using one H200 GPU, MatterLabs Boojum 2.0 prover)

Much commercial and research effort

A (partial) map of companies using and building SNARKs



source: ZKV

Strong demand from industry for ever faster provers

Why so much interest in SNARKs now?

The breakthrough: new SNARK systems with a fast prover

- **Many** commercial applications
- **Many** beautiful ideas

a large bibliography: a16zcrypto.com/zero-knowledge-canon

Applications: (1) Scaling Blockchains

Babai-Fortnow-Levin-Szegedy 1991:

a slow and expensive computer

In this setup, a ~~single reliable PC~~ can monitor the operation of a herd of ~~supercomputers~~ working with unreliable software.

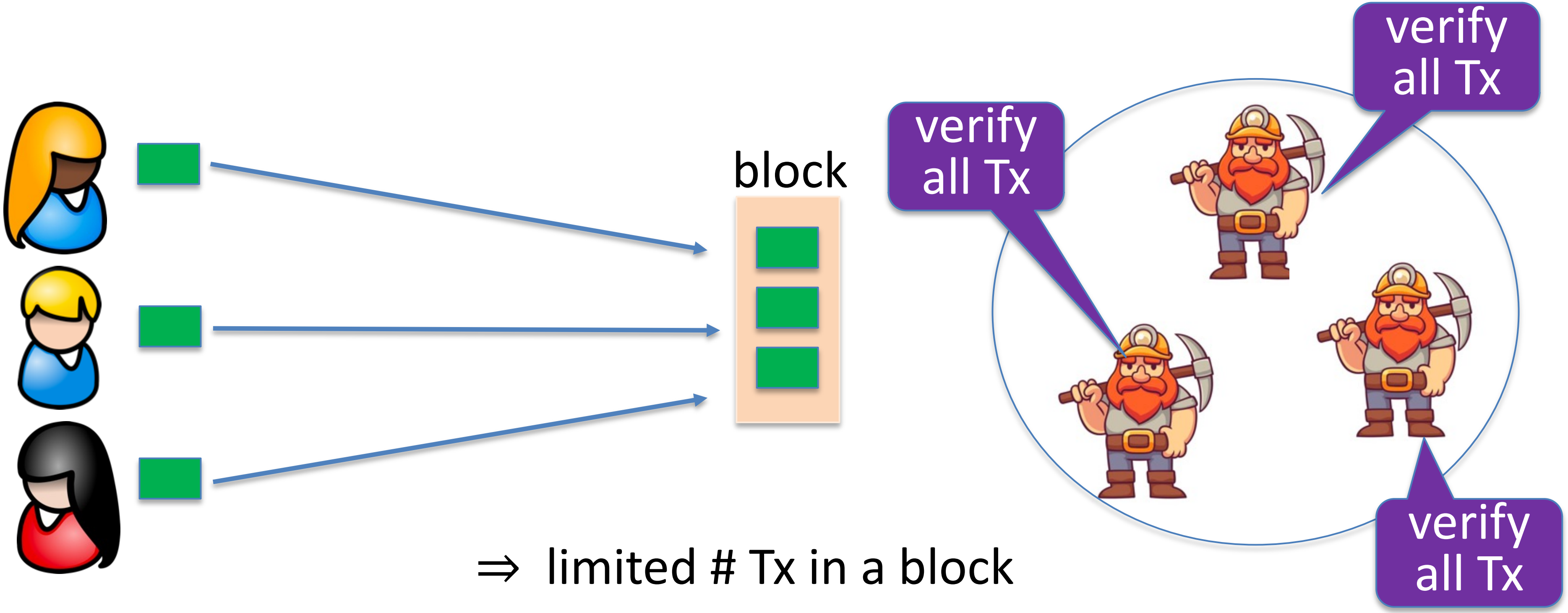
CPUs

“Checking Computations in Polylogarithmic Time”

Applications: (1) Scaling Blockchains

(simplified)

On an L1 chain: every validator verifies all transactions



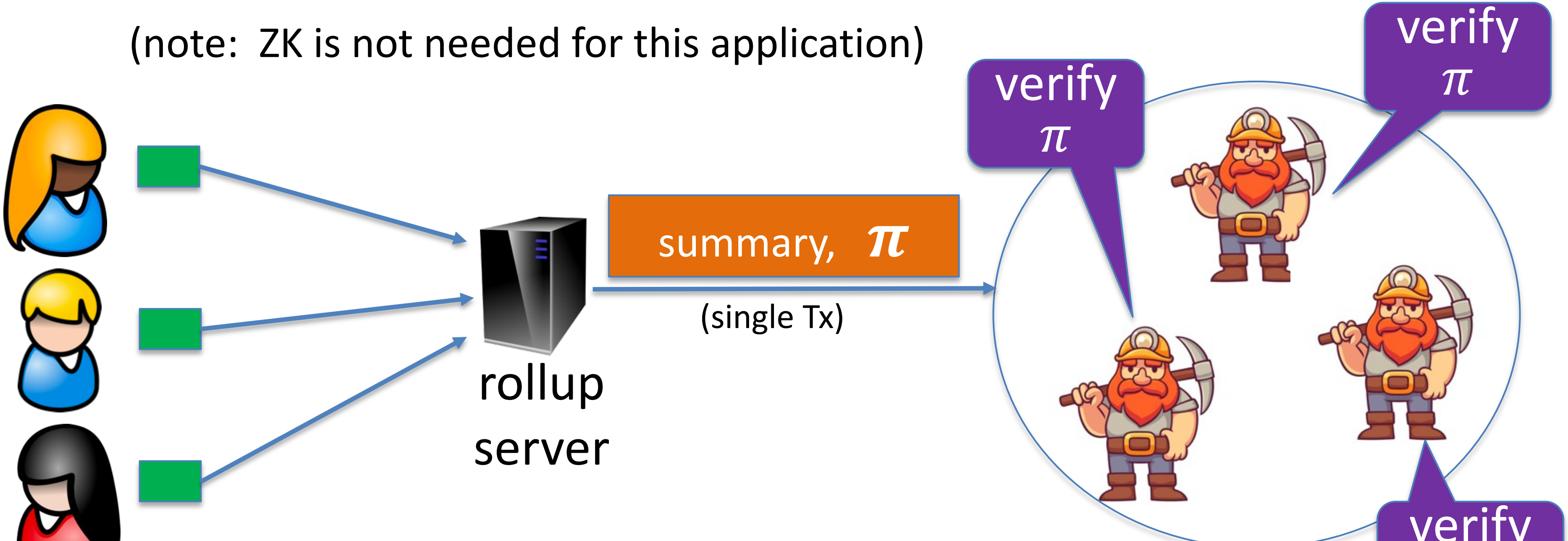
⇒ limited # Tx in a block

Applications: (1) Scaling Blockchains

(simplified)

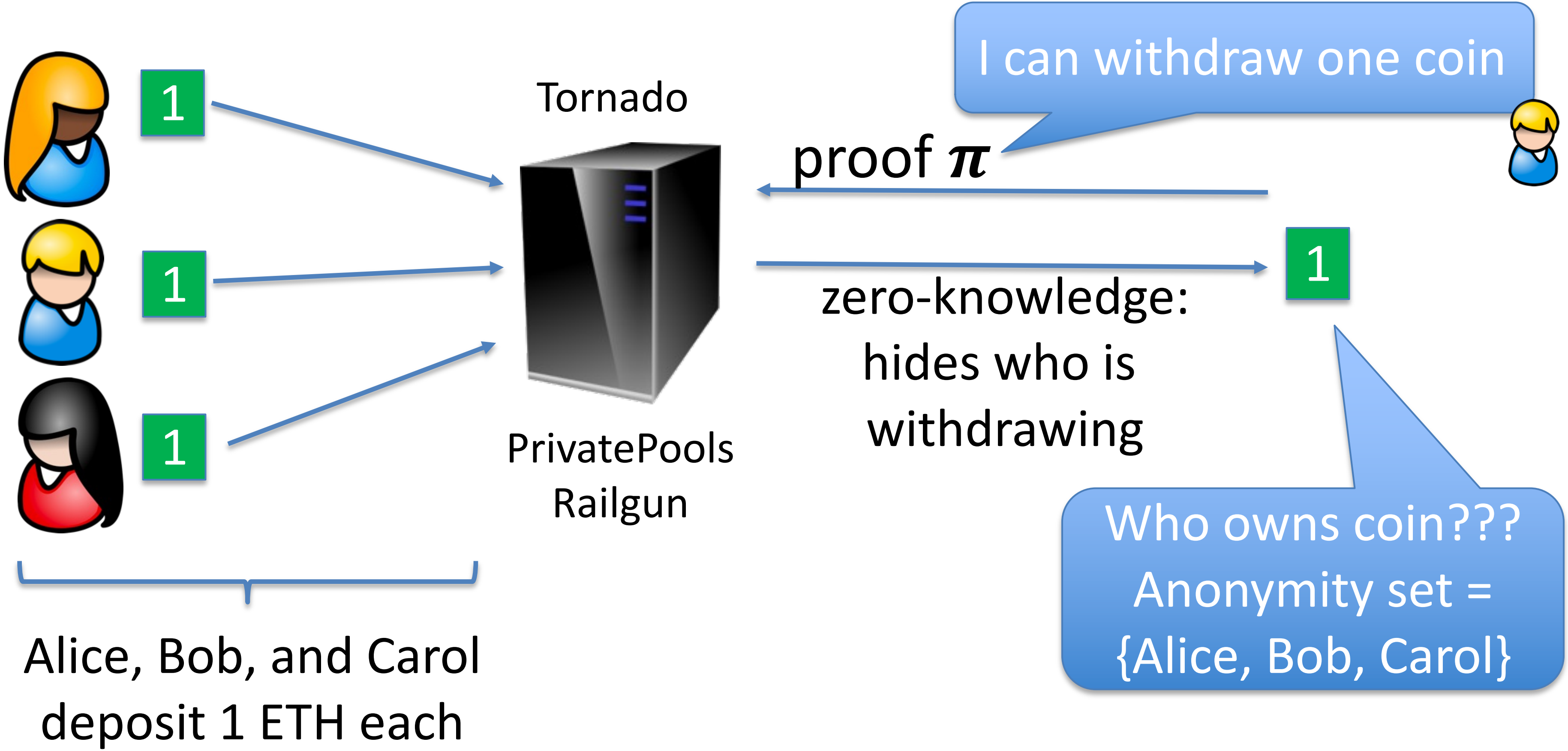
A zk-Rollup: validators only check proof that Tx are valid

(note: ZK is not needed for this application)

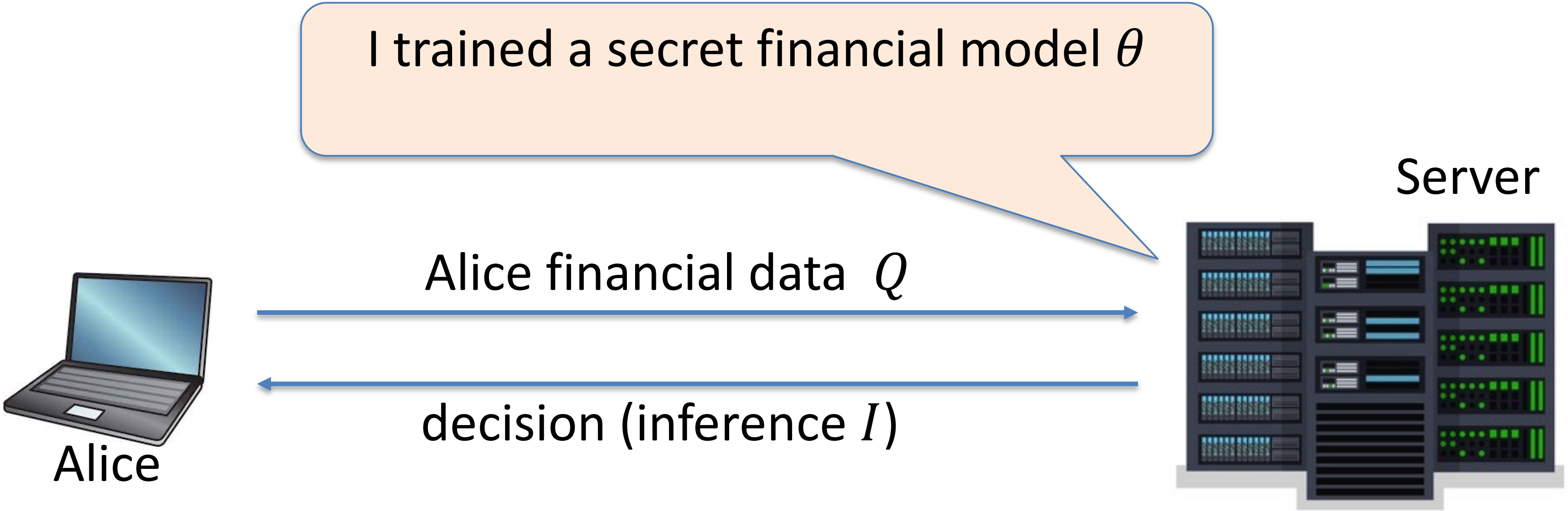


$\Rightarrow$ 100x more transactions per second

Application (1’): private transactions

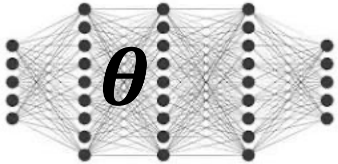


Applications: (2) SNARKs in ML (ZKML)



Is the same model used for everyone ??

Did the server run the model correctly ??



Applications: (2) SNARKs in ML (ZKML)

I trained a secret financial model θ
com $\leftarrow$ Commit(θ)

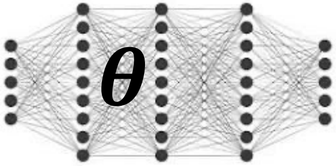
Server

com

Alice

Alice financial data Q →

← decision (inference I), **proof $\boldsymbol{\pi}$**



π proves: server knows $\boldsymbol{\theta}$ s.t.

(i) $f_{\boldsymbol{\theta}}(Q) = I$ and (ii) com = Commit($\boldsymbol{\theta}$)

Applications: (2) SNARKs in ML (ZKML)

I trained a secret financial model θ
com $\leftarrow$ Commit(θ)

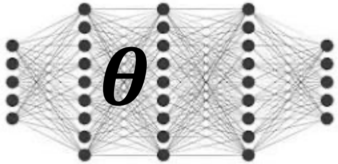
Server

com

Alice

Alice financial data Q

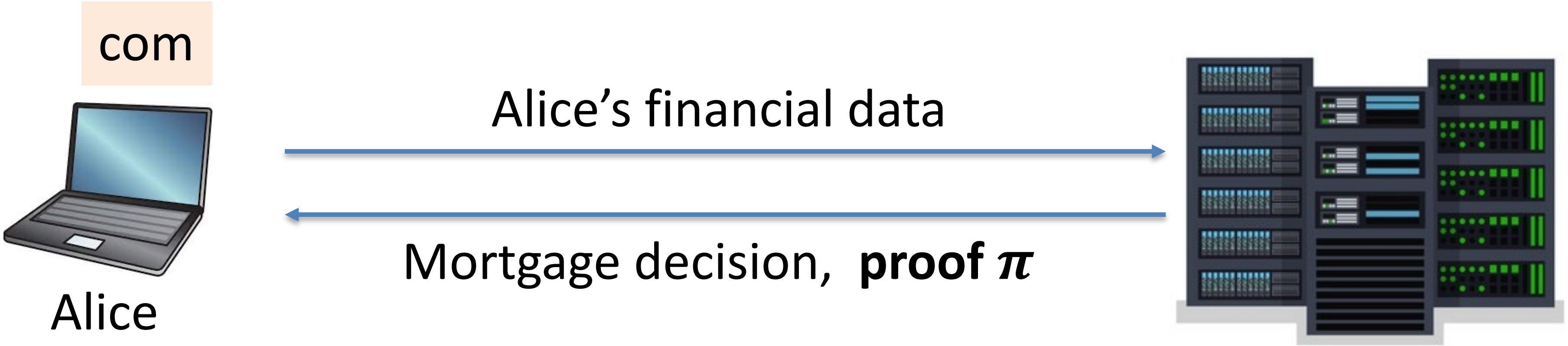
decision (inference I), **proof $\boldsymbol{\pi}$**



Is this practical? Commercial library: EZKL

FairProof: proving model fairness in ZK

[Yadav-Chowdhury-B-Chaudhuri, ICLR’24]

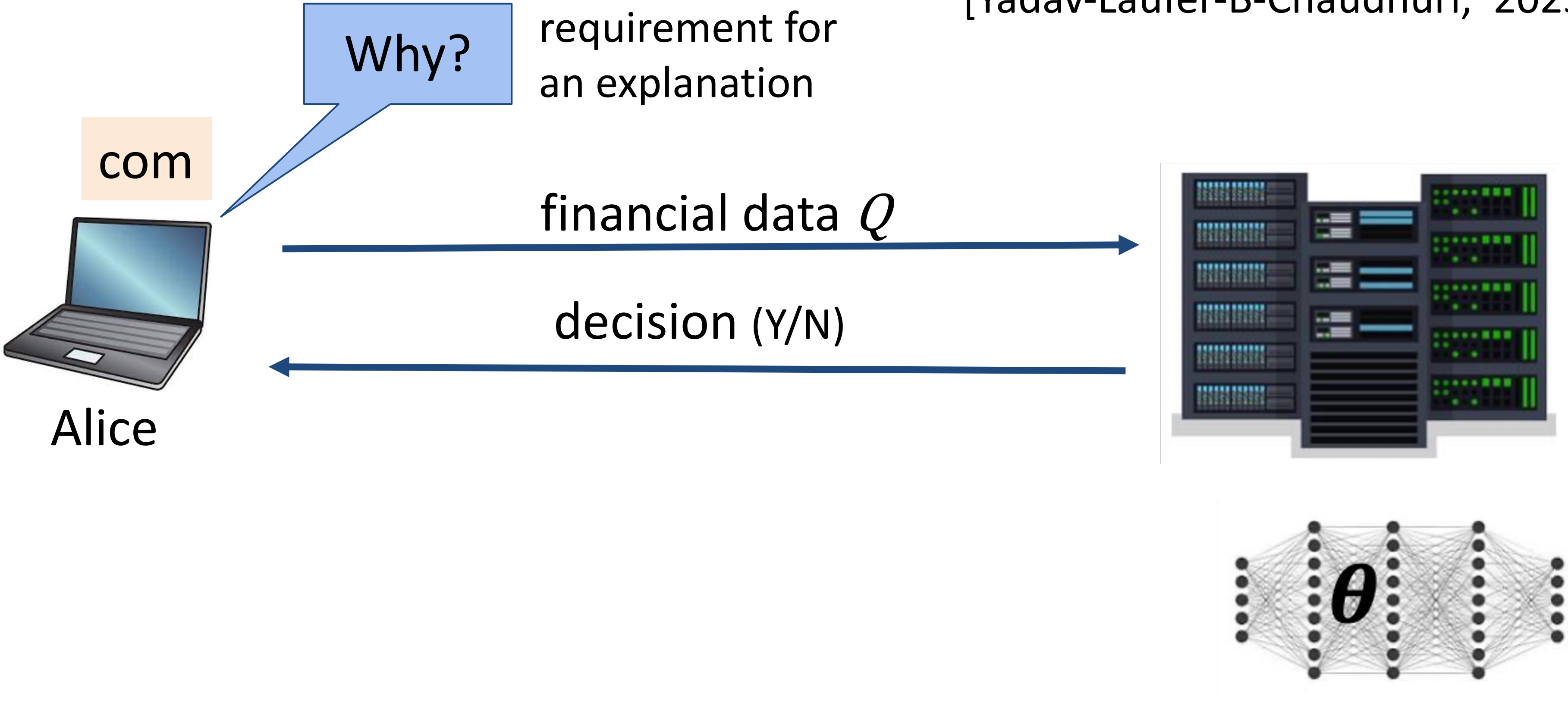


Proof π proves:

- Local Individual Fairness: treating similar people similarly [DHPRZ’12]
- Same model used for everyone

ExpProof: proving AI model explanation in ZK

[Yadav-Laufer-B-Chaudhuri, 2025]



ExpProof: proving AI model explanation in ZK

[Yadav-Laufer-B-Chaudhuri, 2025]

Why?

com

Alice

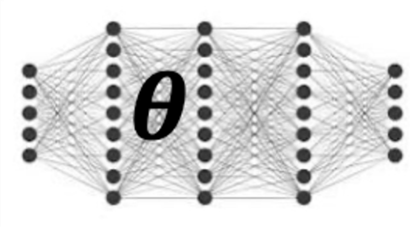
financial data Q

decision D, explanation E, proof π

(LIME)

Proof π:

$$\mathrm{commit}(\Theta) = \mathrm{com}, \quad f_\Theta(Q) = D, \quad \mathrm{LIME}(\Theta, Q) = E$$



Applications: (3) image provenance

These look like prizewinning photos. They're AI fakes.

Artificially generated images of real-world news events proliferate on stock image sites, blurring truth and fiction

By Will Oremus and Pranshu Verma
November 23, 2023 at 6:00 a.m. EST

C2PA: a standard for content provenance

Leica camera has built-in defense against misleading AI, costs $9,125

(also Sony and Nikon)

60MP 

embedded certified signing key ***sk***

location
timestamp
signature

C2PA

verify metadata by checking sig

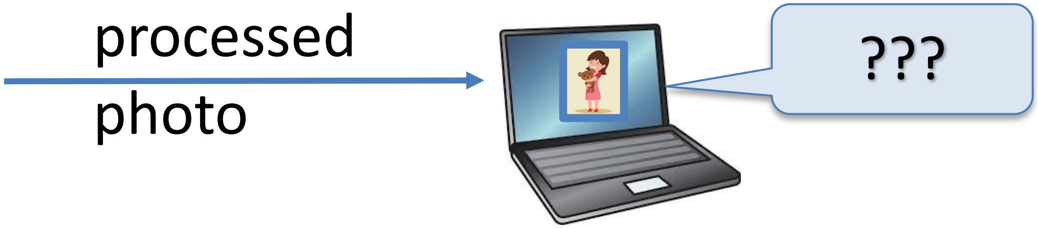
2025: Cloudflare support

A problem: post-processing (editing)

Newspapers often process the photos before publishing:

- Resize (1500 × 1000), Crop, Grayscale, Blur face (AP lists allowed ops)

The problem: laptop cannot verify signature on processed photo



The Solution proposed by C2PA is not ideal … is there a better solution?

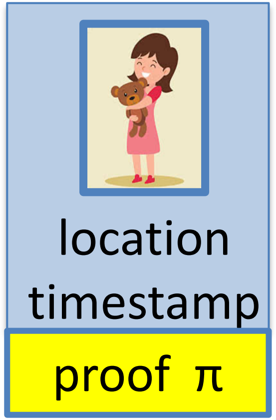
A Cryptographic Solution: zkSNARKs

public statement

Laptop has (***Edited***, ***Ops***). Editing software attaches a proof π that:

> I know a witness (***Orig, Sig***) such that
>
> 1. ***Sig*** is a valid C2PA signature on ***Orig***
> 2. ***Edited*** is the result of applying ***Ops*** to ***Orig***
> 3. metadata(***Edited***) = metadata(***Orig***)

edited photo



$\Rightarrow$ Laptop verifies π and shows metadata to user

Application (4): liberating Web data

Goal: ZK proof that Bob's bank account balance > X

ZK proof that Bob bought a ticket to the Lakers game

ZK proof that Bob ordered DoorDash 10 times this month

• • •

The challenge: no changes to web site!

zk TLS (DECO: CCS'2020)

The problem: TLS payload is not authenticated

browser

session-key

TLS session setup

(signed by web server)

TLS HTML payload

(encrypted with session key)

HTTPS
web server

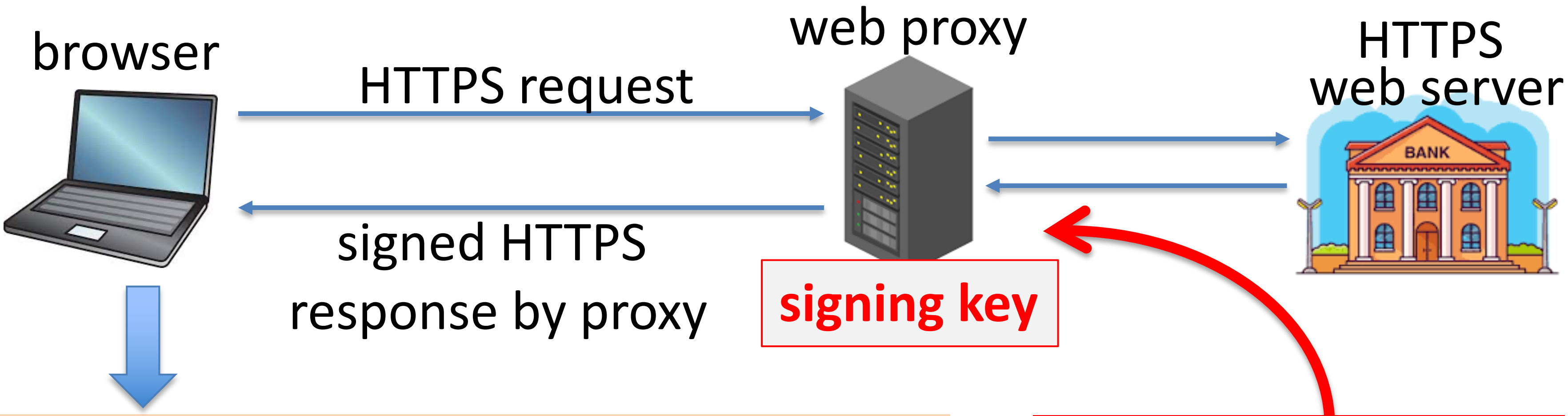
session-key

⇒ enc. payload can be forged by client

Future: RFC 9421 (HTTP msg sigs)

A TLS Proxy Design

(simplified)



Browser generates ZK proof that:

- HTTPS handshake is signed by bank
- encrypted payload is signed by proxy
- decrypted payload says balance > X

A network attack:
cause proxy to sign incorrect encrypted TLS frame

Course organization

1. Next lecture: what is a succinct ZK proof? (definitions)
2. Bommer ZK proofs: $\Sigma-$protocols and their applications
3. First succinct proofs: Bulletproofs and Groth16
4. Succinct proof toolchains
5. Modern succinct proof systems:

   Plonk, HyperPlonk, code-based proofs
6. SNARK recursion and folding: reducing memory needs

Course organization

cs355.stanford.edu

- Homework problems and project. No final exam.
- Optional weekly sections on Friday

Please tell us how we can improve …
Don’t wait until the end of the quarter

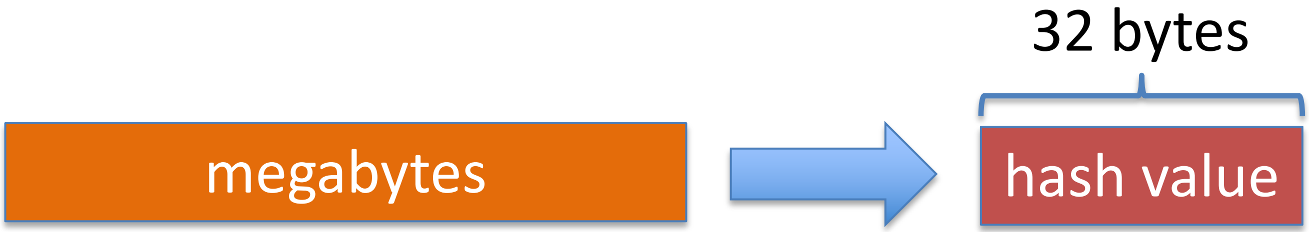
Let’s get started …

Cryptography Background

(1) cryptographic hash functions

An efficiently computable function $H: M \to T$

where $|M| \gg |T|$



$T = \{0,1\}^{256}$

Collision resistance

<u>Def</u>: a **<u>collision</u>** for $H: M \to T$ is pair $x \neq y \in M$ s.t. $H(x) = H(y)$

$|M| \gg |T|$ implies that <u>many</u> collisions exist

<u>Def:</u> a function $H: M \to T$ is **<u>collision resistant</u>** if it is "hard" to find even a single collision for H (we say H is a CRH)

Example: **SHA256**: $\{x : \text{len}(x) < 2^{64} \text{ bytes}\} \to \{0,1\}^{256}$

(output is 32 bytes)

details in CS255

(2) Cryptographic Commitments

<u>Def</u>: a **commitment scheme** is a pair of eff. algorithms (C, V) where

- $C(m, r) \to com$ commits to a message $m \in \mathcal{M}$ using randomness $r \in \mathcal{R}$
- $V(m, r, com) \to 0/1$

such that for all $m \in \mathcal{M}, r \in \mathcal{R}$: $V(m, r, C(m, r)) = 1$.

The scheme is **computationally binding** if for every efficient adv. A:

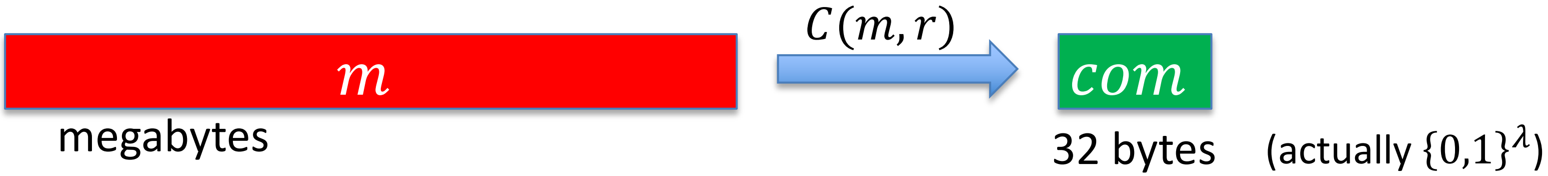
$$\Pr\big[\mathcal{A}() \to (com, m_0, r_0, m_1, r_1) \;:\; V(m_0, r_0, com) = V(m_1, r_1, com) = 1\big] < negl()$$

The scheme is **unconditionally hiding** if for every adv. A and all $m_0, m_1 \in \mathcal{M}$

$$\Big|\Pr\big[\mathcal{A}(C(m_0, r_0)) = 1\big] - \Pr\big[\mathcal{A}(C(m_1, r_1)) = 1\big]\Big| < negl(), \quad \text{where } r_0, r_1 \leftarrow \mathcal{R}$$

(2) Cryptographic Commitments

Def: a commitment scheme (C, V) is **succinct** if the size of com is independent of the size of m



Note: an unconditionally binding commitment scheme cannot be succinct. Why?

Def: a **binding commitment scheme** is a commitment scheme that is binding but not necessarily hiding.

A commitment scheme from a CRH

Let $H: \mathcal{M} \times \mathcal{R} \to T$ be a hash function

Define: $C(m,r) \coloneqq H(m,r)$ and $V(m,r,com) = 1$ iff $H(m,r) = com$

<u>Thm 1</u>: if H is CRH then (C,V) is a computationally binding scheme

<u>Thm 2</u>: if for all $m \in \mathcal{M}$ the distr. $\{H(m,r) : r \leftarrow \mathcal{R}\}$ is uniform in T then (C,V) is an unconditionally hiding scheme

Note: when $T = \{0,1\}^\lambda$ the commitment scheme is succinct

(3) Vector commitments

Def: a **vector commitment scheme** is a triple of eff. algorithms (C, O, V) s.t.

- $C(v, r) \to com$ commits to a vector $v \in \boldsymbol{W}^n$ using randomness $r \in \mathcal{R}$
- $O(v, r, i) \to \pi$ for $i \in [n]$ outputs a proof π for the value of $v[i]$
- $V(com, u \in \boldsymbol{W}, i, \pi) \to 0/1$ verifies that π is a valid proof that $v[i] = u$

such that for all $v \in \boldsymbol{W}^n, r \in \mathcal{R}, i \in [n]$: $V(\, C(v,r),\ v[i],\ i,\ O(v,r,i)\,) = 1$.

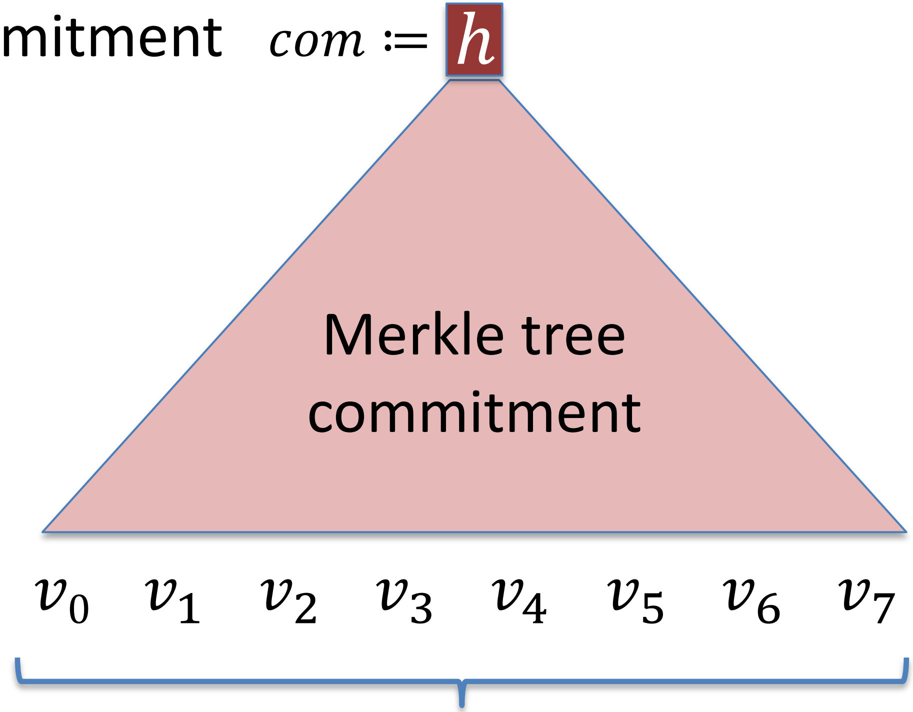
Def: the scheme is **binding** for $n \in \mathbb{N}$ if for every efficient adv. A:

$$\Pr\left[\mathcal{A}() \to (com, i \in [n], u_0, \pi_0, u_1, \pi_1) \ : \ \begin{array}{c} V(com, u_0, i, \pi_0) = V(com, u_1, i, \pi_1) = 1 \\ \text{and} \quad u_0 \neq u_1 \end{array}\right] < negl()$$

Hiding (informally): defined as for commitments, but holds for all unopened cells, after adversary sees a bunch of opening proofs chosen by the adversary.

Merkle tree (Merkle 1989)

commitment $com := h$



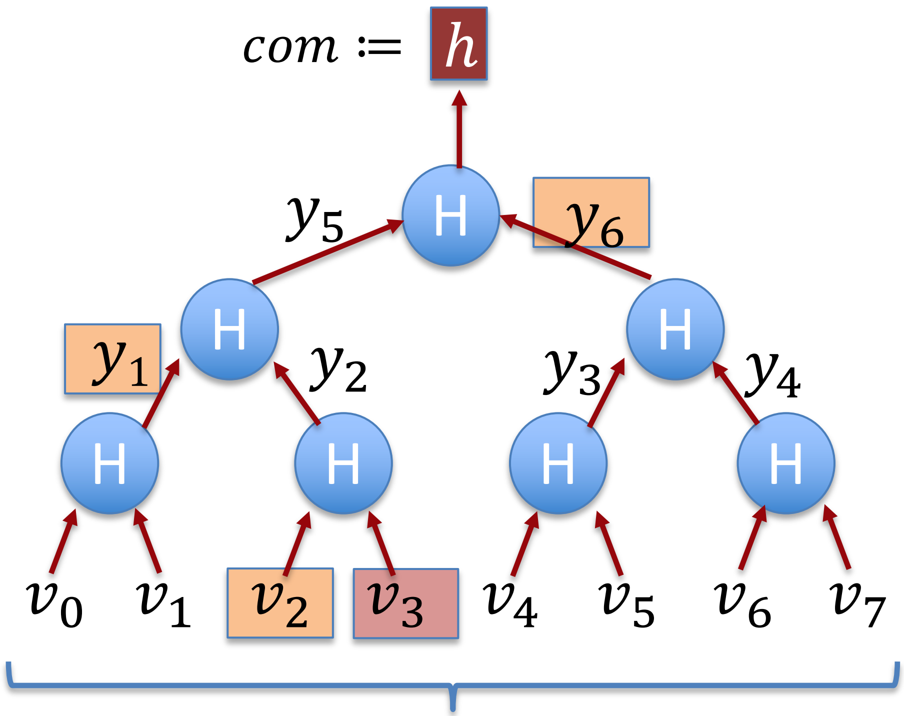
a vector $v \in V^8$

Goal:

- commit to a vector v
- Later prove $v[i] = u$

Merkle tree (Merkle 1989) [simplified]

$H: X^2 \to X$



a vector $v \in V^8$

Goal:

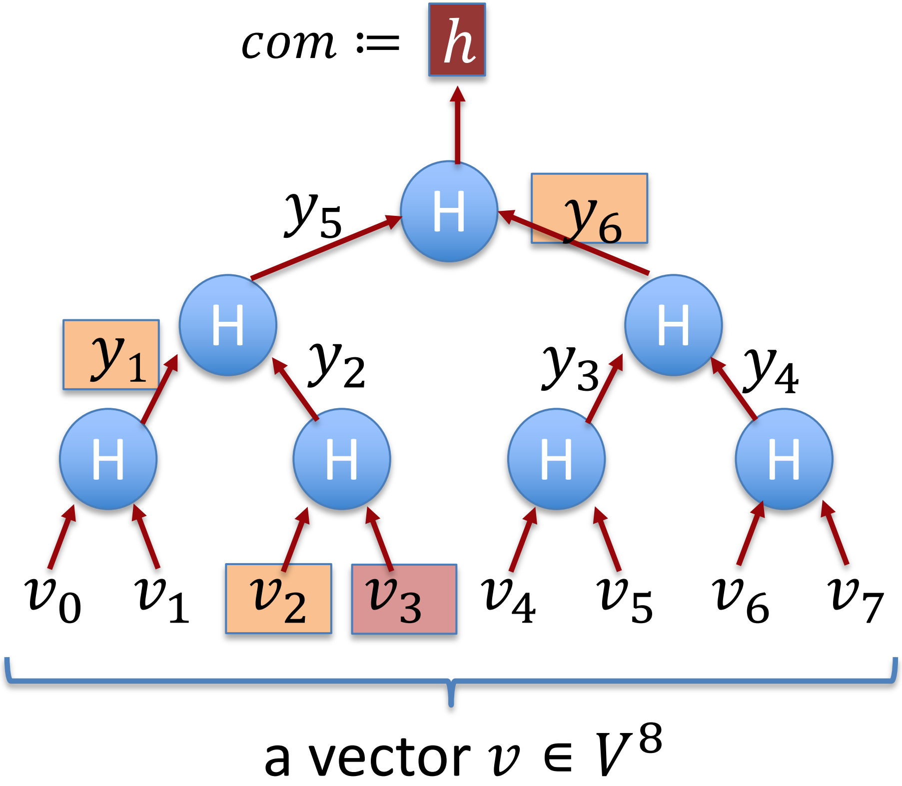
- commit to a vector v
- Later prove $v[i] = u$

To prove $v[3] = u$,

proof $\pi := (v_2, y_1, y_6)$

length of proof: $\log_2 n$

Merkle tree (Merkle 1989) [simplified]



To prove $v[3] = u$,

proof $\pi := (v_2, y_1, y_6)$

Alg. $V(com, u, i = 3, \pi)$:

$y_2 \leftarrow H(v_2, u)$

$y_5 \leftarrow H(y_1, y_2)$

$h' \leftarrow H(y_5, y_6)$

accept if $h' = com$

Merkle tree (Merkle 1989)

<u>Thm</u>: if H is a CRH then Merkle is a binding vector commitment for all bounded (poly-size) n.

We will use this a lot !!

Question: how to make this hiding?

END OF LECTURE

Next lecture: definitions and a first example