Succinct Arguments

Lecture 01: Introduction and Background

Pratyush Mishra
UPenn
Fall 2025

Logistics

- **Time:** Mondays and Wednesdays, 10:15AM-11:45AM
- **Location:** AGH 214
- **Course Website:** pratyushmishra.com/classes/cis-7000-f25/
- **Canvas: TBD**
  - Reading assignments will be posted here
- **EdStem: TBD**
  - We'll use this for all course communications
  - Ask and answer questions!
- **Waitlist:** email me (**prat@seas.upenn.edu**) after class

Grading

Four key components to grading:

- **Attendance + Participation (15%)**
  - This is a research seminar! We're here to learn by discussing papers, and that requires participation.
  - Can also participate on Ed (eg: asking + answering questions)
- **Reading assignments (15%)**
  - For classes marked as discussions, I will post an short-answer assignment on Canvas before-hand
- **Leading a paper discussion (25%)**
  - Students are expected to lead a discussion on a paper. This will likely happen in the 2nd half of the class
- **Final Project (45%)**
  - Research project/literature survey

Course Structure

LECTURE

Part I: Theory

- What are interactive proofs and ZKPs?
- What is a zkSNARK?
- Constructions of zkSNARKs for circuits
  - From Linear IPs
  - From Polynomial IOPs + various polynomial commitments
- Recursive composition of SNARKs

Part II: Programming SNARKs

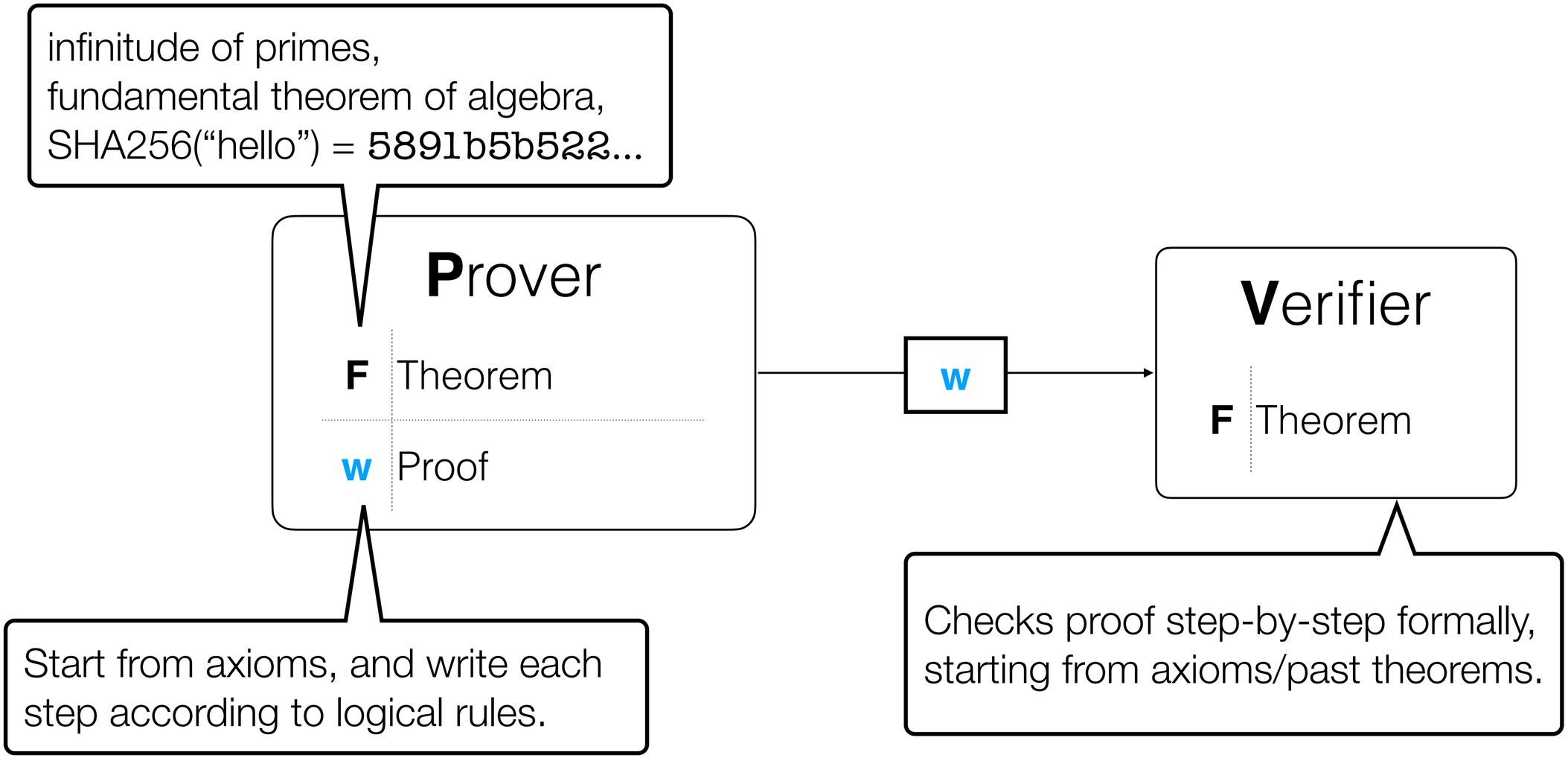
- PLs for SNARKs
- Formal verification for SNARKs
- Implementation/Systems for SNARKs

Part III: Applications

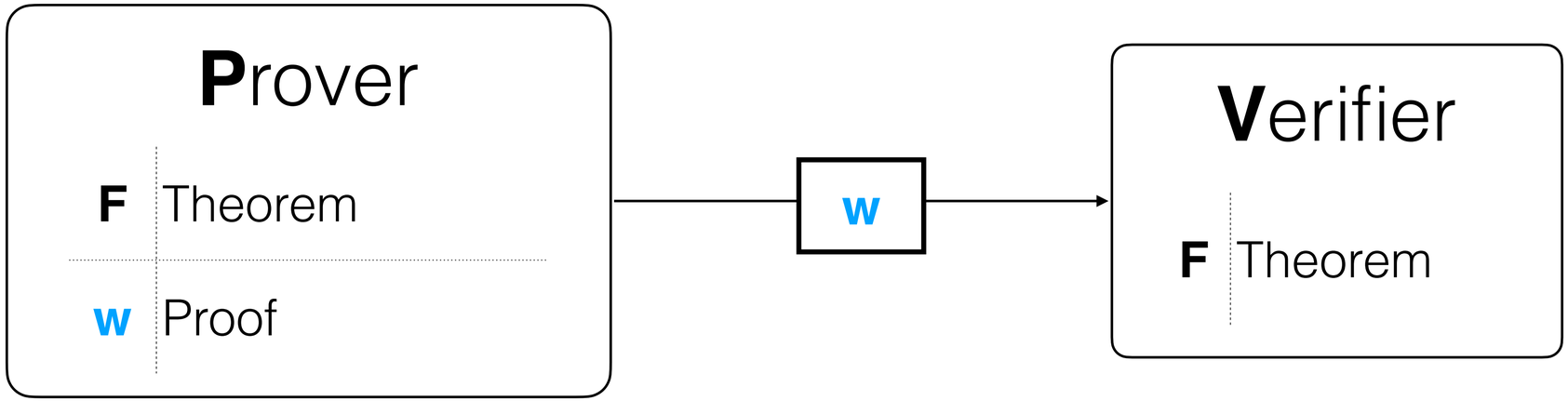
- Blockchains/transparency logs:
  - Privacy-preserving payments
  - Privacy-preserving smart-contracts
  - Rollups
  - Succinct blockchains
- Anonymous authentication/credentials
- Collaborative proving
- SNARKs to authenticate images/text/video (stop ChatGPT!)

DISCUSSION

What does it mean to prove something?



Mathematical proofs = NP



Completeness: For all *true* theorems, ∃ a proof **w** that convinces the verifier

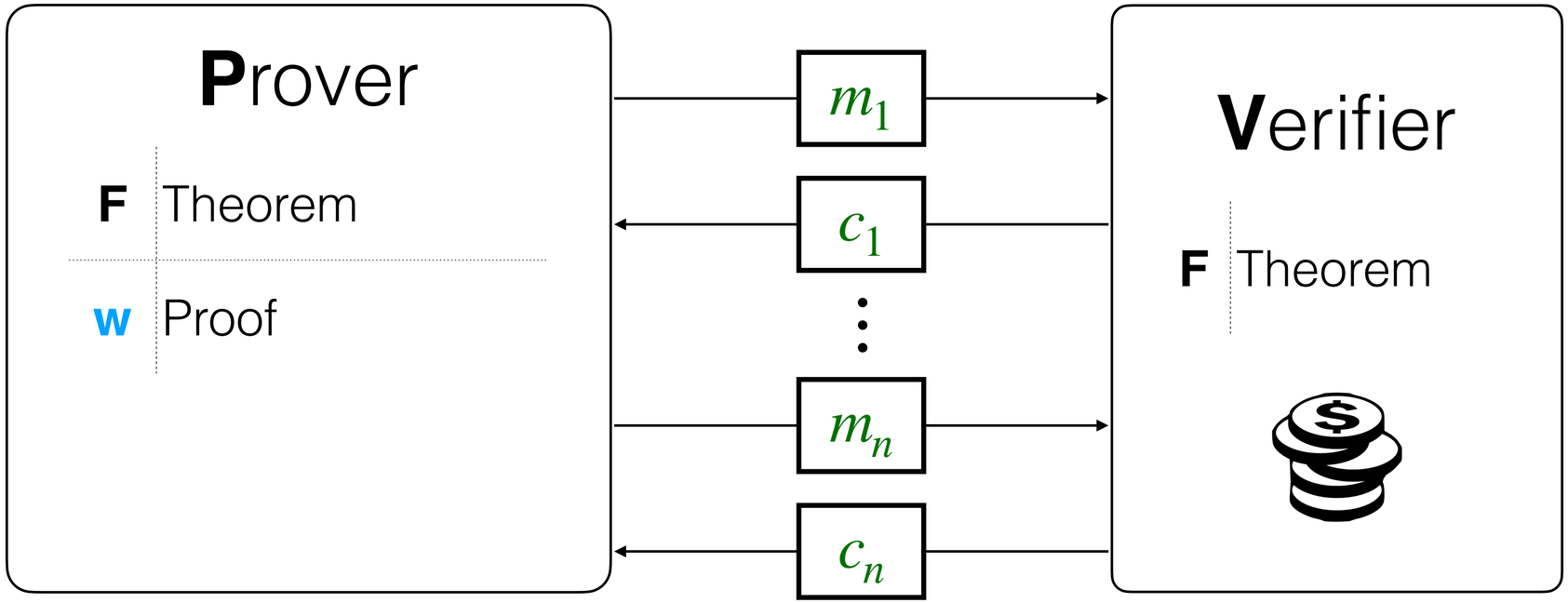
Soundness: For all *false* theorems, no claimed proof **w** can convince the verifier

Efficiency: The verifier is *deterministic* and *runs in polynomial time.*

II

NP

Adding randomness and interaction



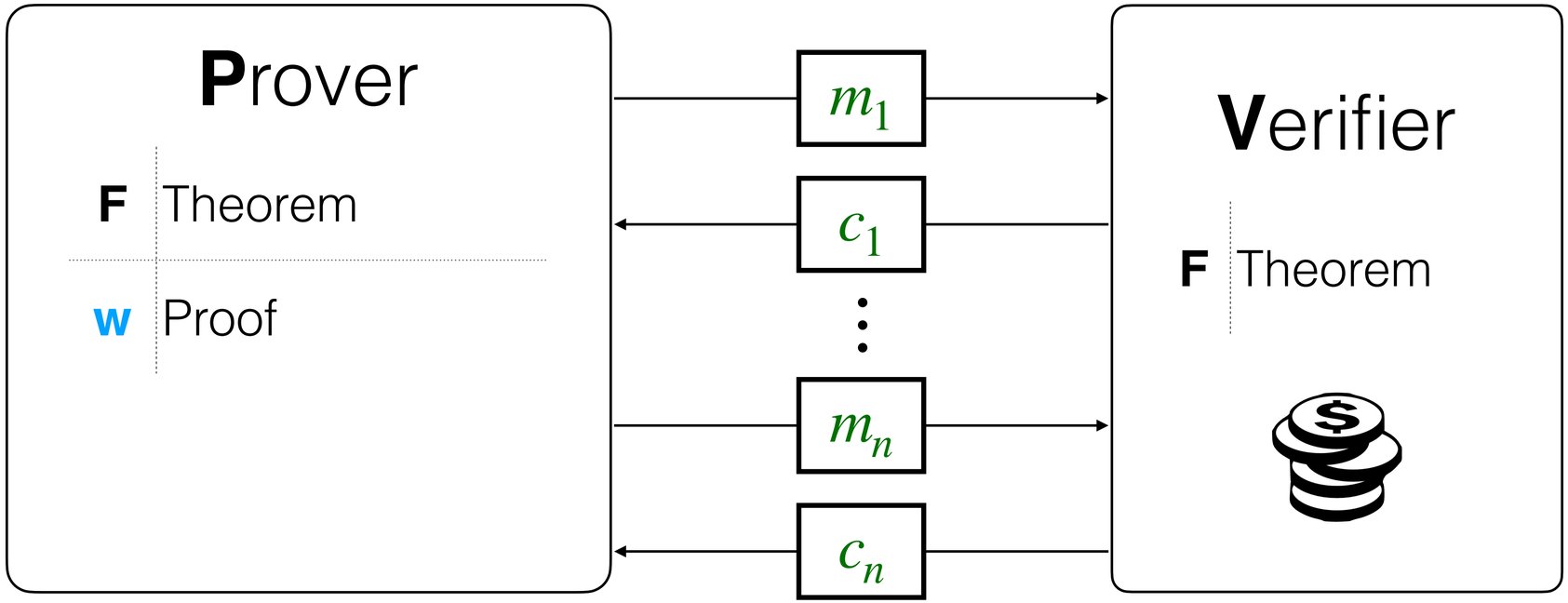
Completeness: For *true* theorems, ∃ a prover that convinces the verifier wp 1.

Soundness: For *false* theorems, no prover can convince the verifier wp ≥ 1/2.

Efficiency: The verifier is *randomized* and *runs in probabilistic polynomial time.*

= **IP** [GMR85]

Does it help? Yes!

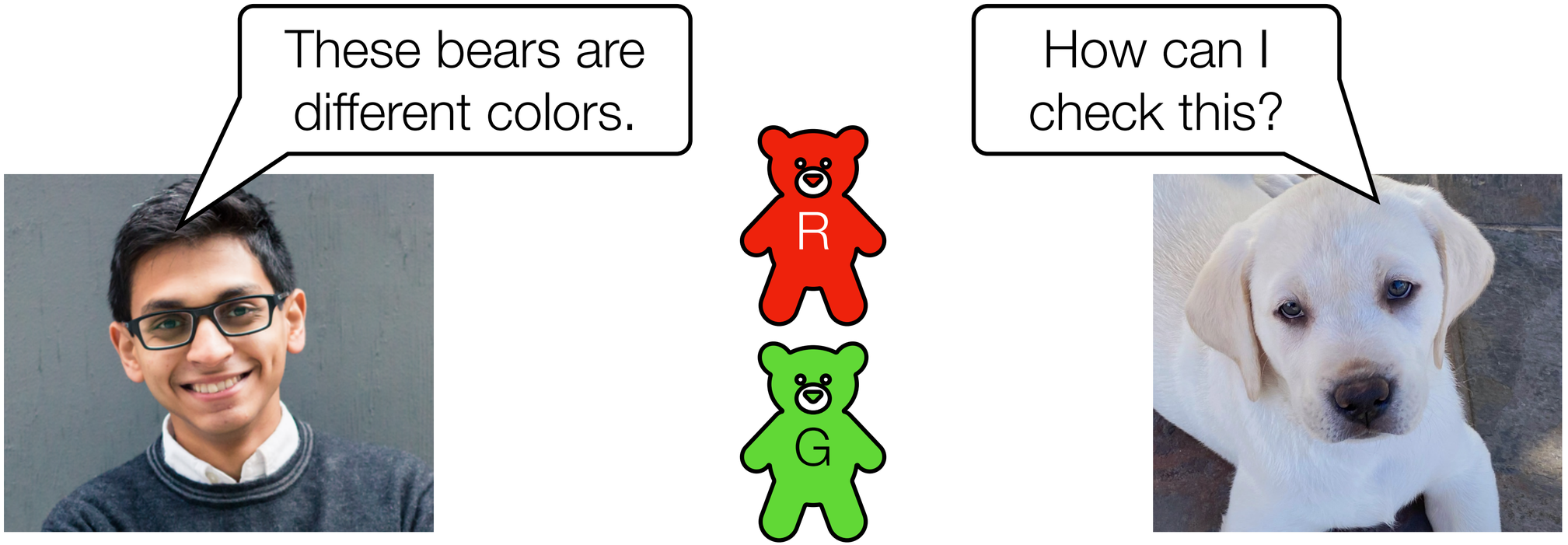


CoNP ⊆ IP [GMW86]

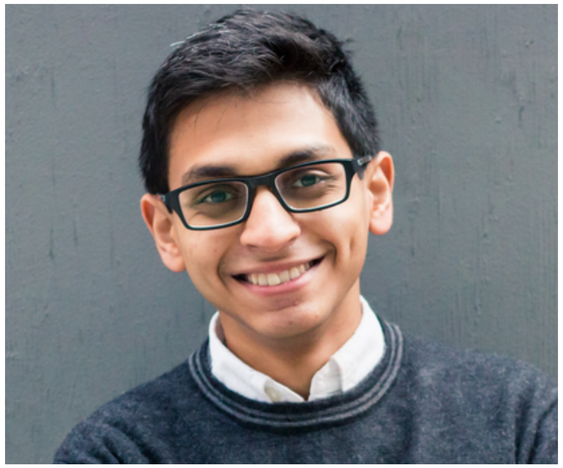
IP = PSPACE [S92]

Delegation of computation [GKR08]

Example: Color-blindness test



Example: Color-blindness test

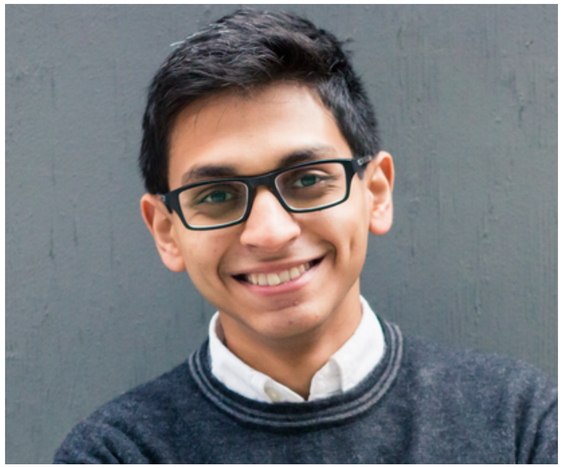


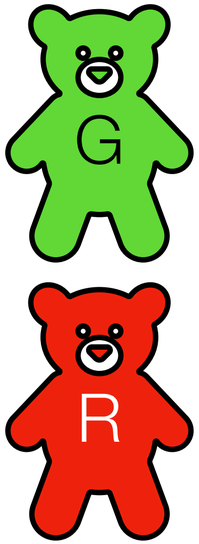
$b \leftarrow \{0,1\}$

If $b = 0$, do nothing

If $b = 1$, shuffle

Example: Color-blindness test





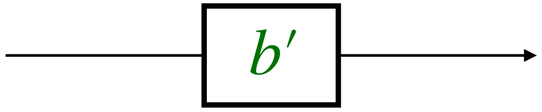
$b \leftarrow \{0,1\}$

If $b = 0$, do nothing

If $b = 1$, shuffle

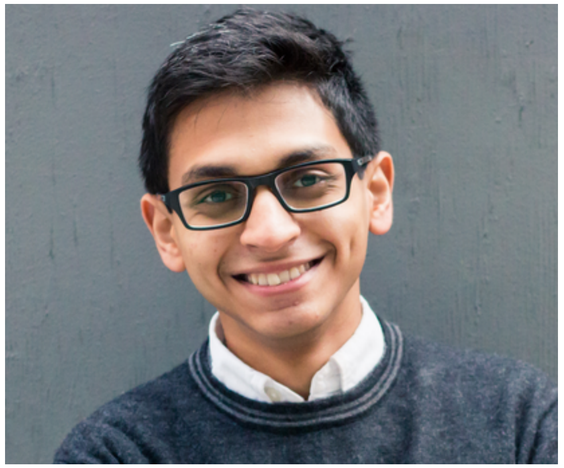
$b' := 1$, if shuffled

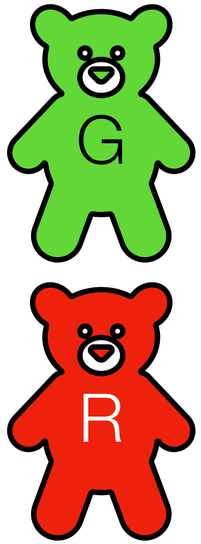
$b' := 0$, if not



$b \stackrel{?}{=} b'$

Example: Color-blindness test





Completeness: If the colors are different, then I will always detect shuffles.

Soundness: If the colors are not different, then I will guess wrong 1/2 the time.

Efficiency: Verifier only needs to flip a coin and shuffle.

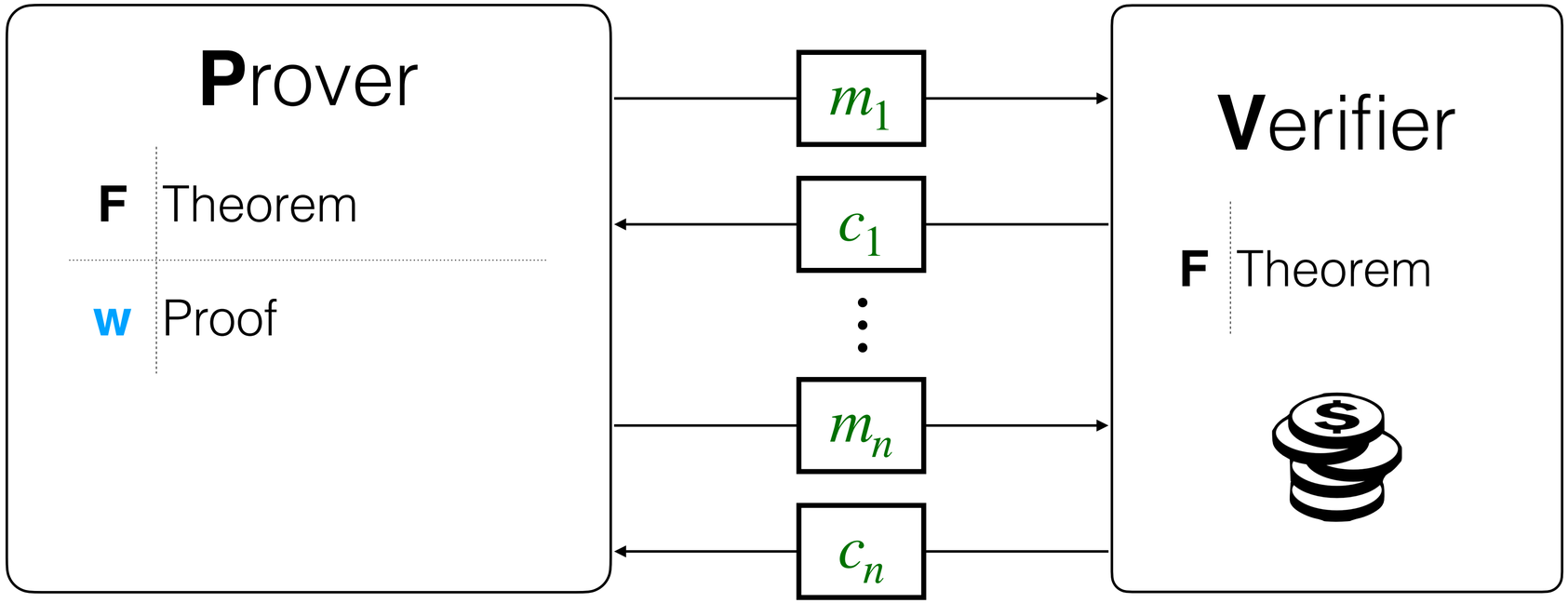
What about privacy?

Let’s say the prover exerted a lot of effort in trying to find the proof of a difficult conjecture.

She wants to get recognition for this, but doesn’t trust others to not steal credit.

She needs a ***zero-knowledge proof.***

Zero-knowledge proofs



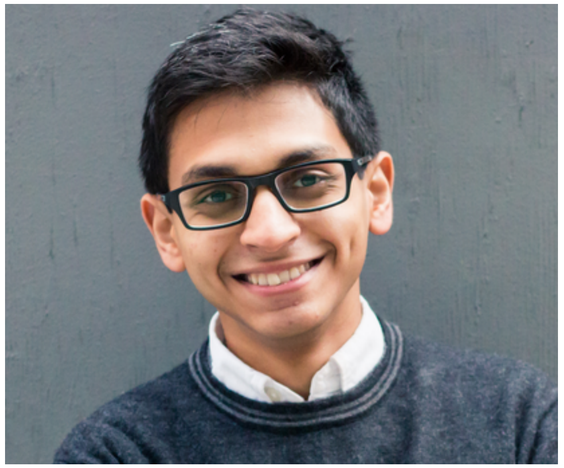
Completeness: For *true* theorems, ∃ a prover that convinces the verifier wp 1.

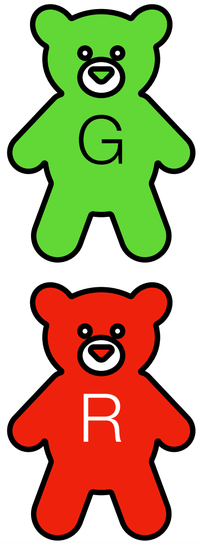
Soundness: For *false* theorems, no prover can convince the verifier wp ≥ 1/2.

Efficiency: The verifier is *randomized* and *runs in probabilistic polynomial time.*

Zero-knowledge: The verifier learns nothing about **w** except that it's valid.

Example: Color-blindness test





Completeness: If the colors are different, then I will always detect shuffles.

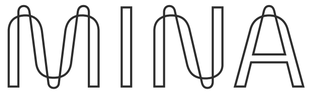
Soundness: If the colors are not different, then I will guess wrong 1/2 the time.

Efficiency: Verifier only needs to flip a coin and shuffle.

Zero-knowledge: The verifier learns only that the colors are different; nothing else!

Many applications!

Private transactions



…

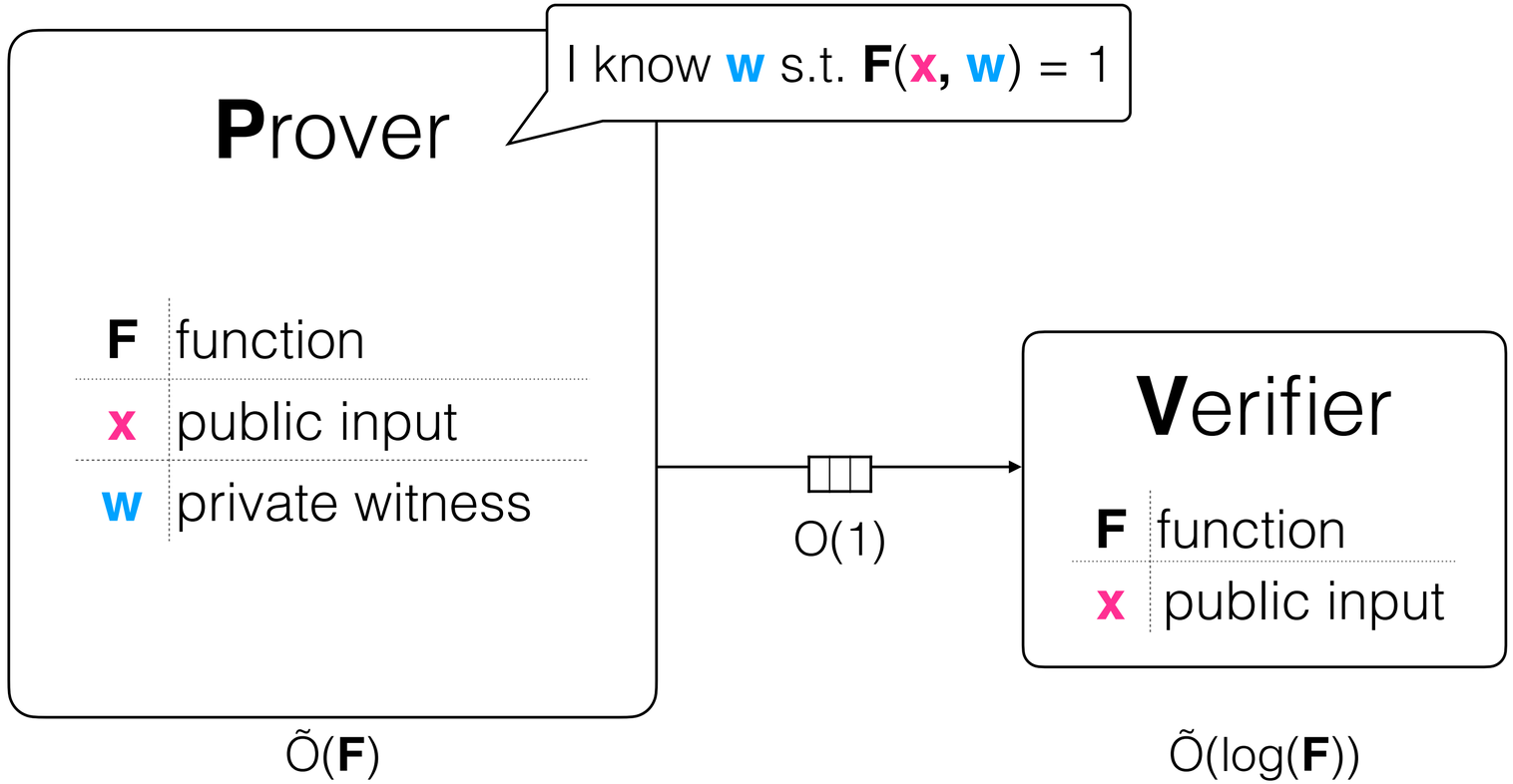
Scalable and/or Private Smart Contracts

Decentralized multiplayer games

- **Anonymous credentials** [DFKP16]
- **Prove existence of security vulnerability** [DARPA Sieve, OBW22]
- **Coercion-resistant voting** [MACI]
- …

Succinct Non-Interactive Arguments (SNARKs)

[Mic94, Groth10, GGPR13, Groth16…
…, GWC19, CHM**M**VW20, …]



Succinctness: V runs in time much less than |**F**|

How to construct zkSNARKs?

A: Polynomials!