THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Department of Computing
電子計算學系

Building SNARKs

(credit to Alessandro Chiesa)

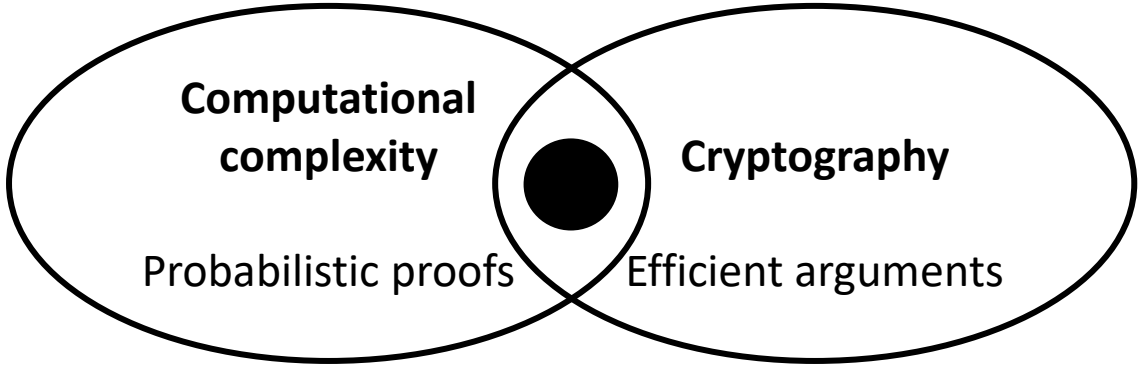
GAO Shang

2025/07/03

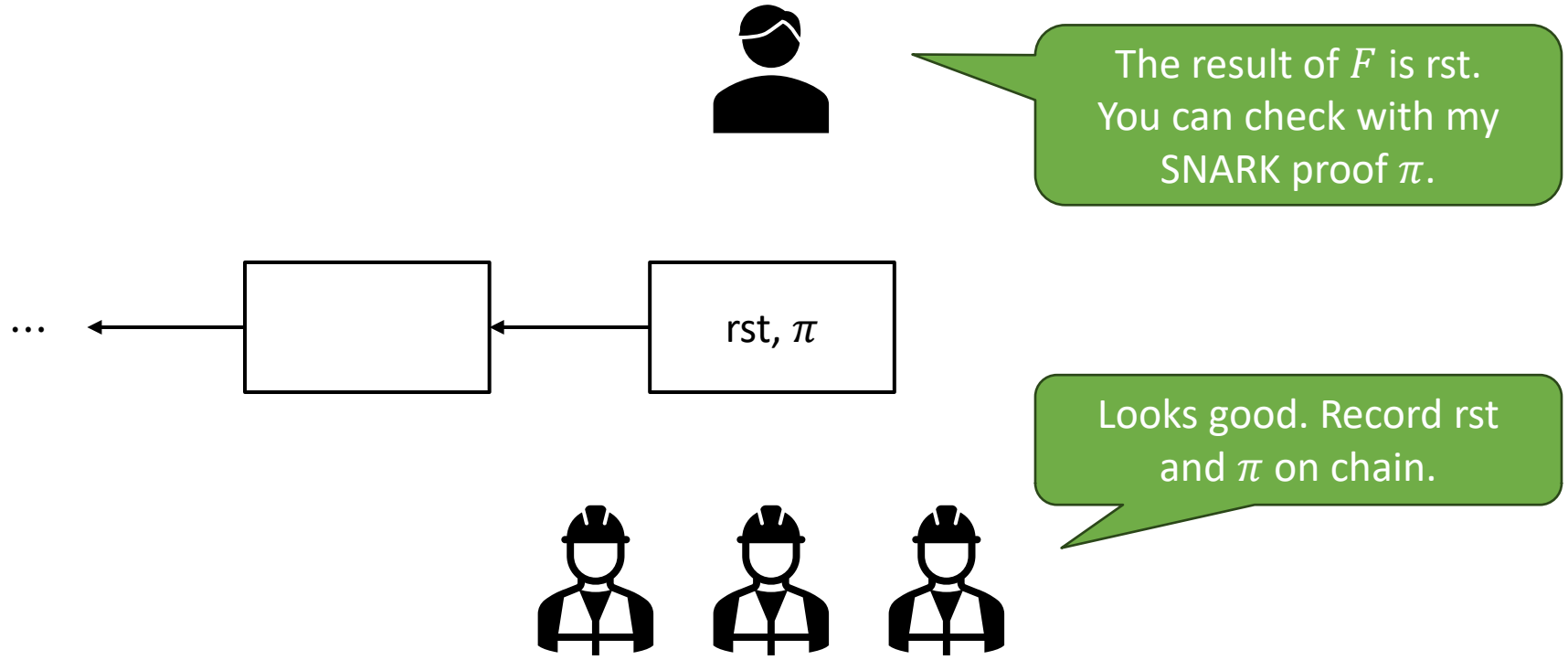
SNARKs

- Cryptographic proofs for computation integrity that are super **short** and are super **fast** to verify.

- Origins in the 1990s:

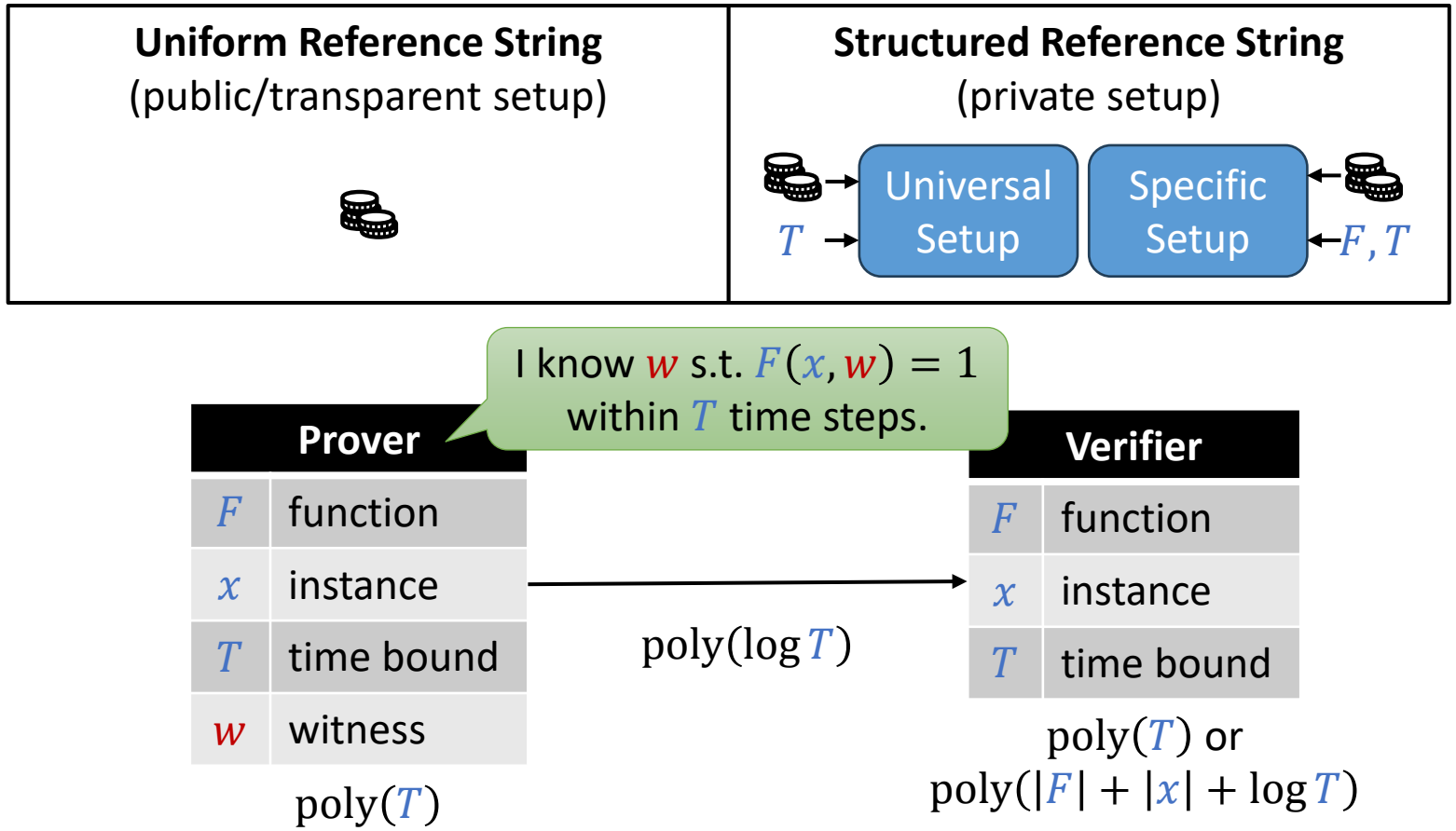


Blockchain Application



SNARGs

- Succinct Non-interactive ARGuments (SNARK: SNARG of Knowledge).



Models

- Different models depending on the "powers" granted to the verifier:

multi-prover
isolated multiple provers

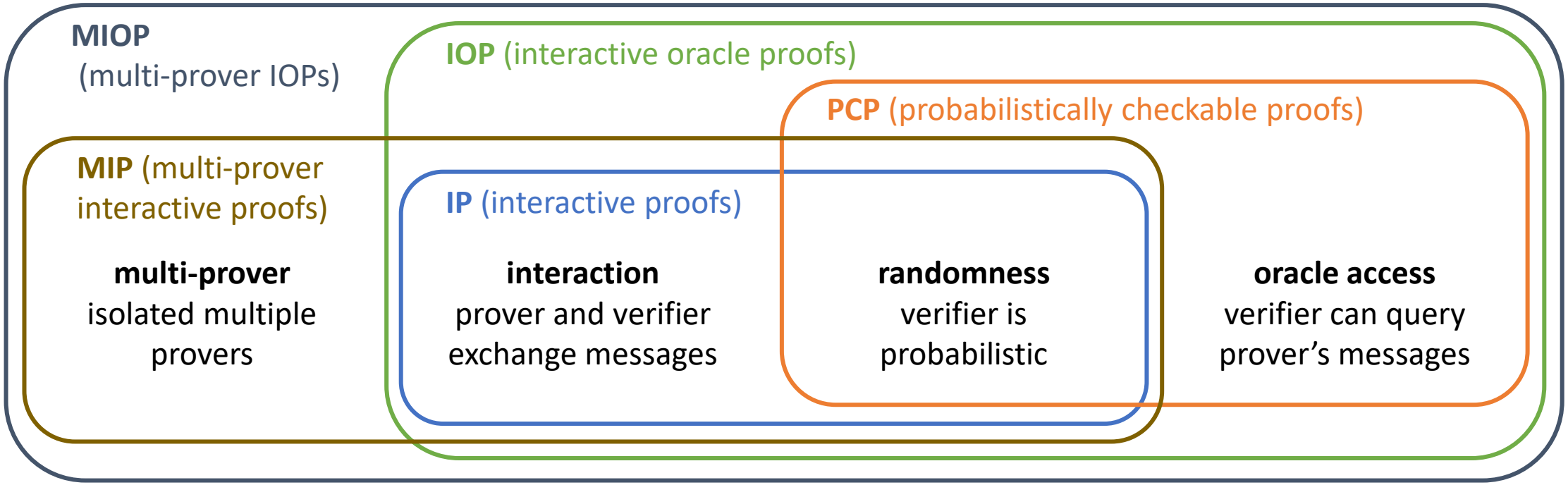
interaction
prover and verifier exchange messages

randomness
verifier is probabilistic

oracle access
verifier can query prover’s messages

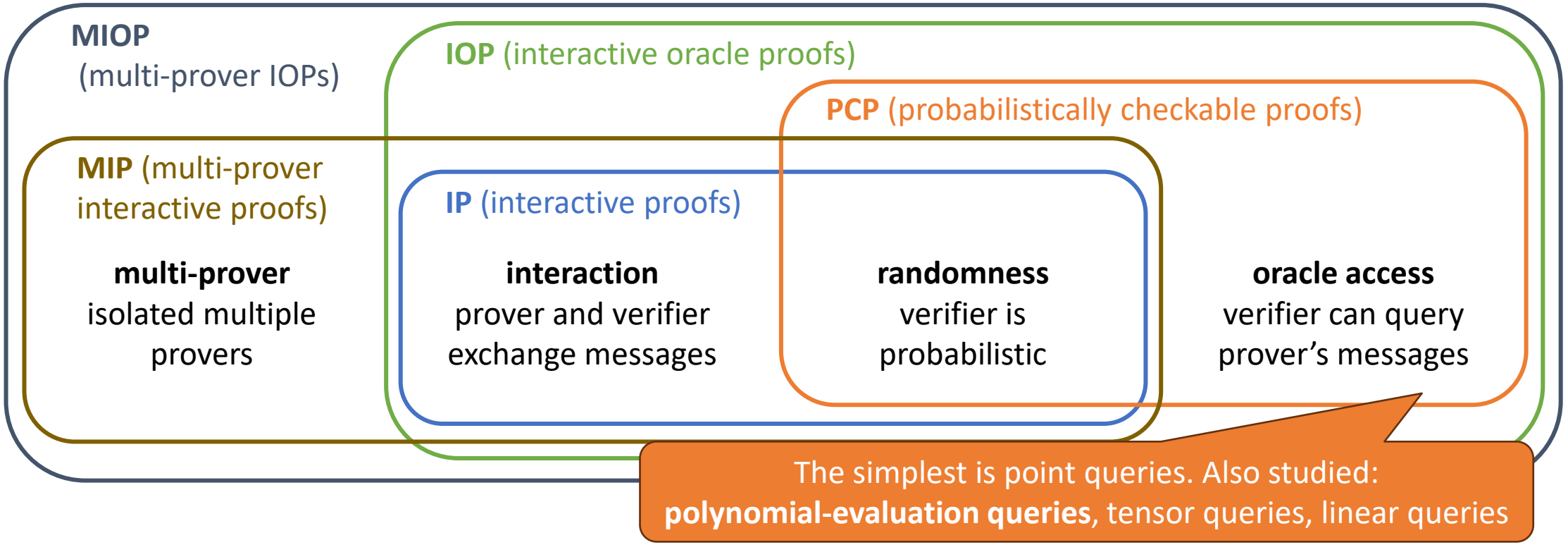
Models

- Different models depending on the "powers" granted to the verifier:



Models

- Different models depending on the "powers" granted to the verifier:

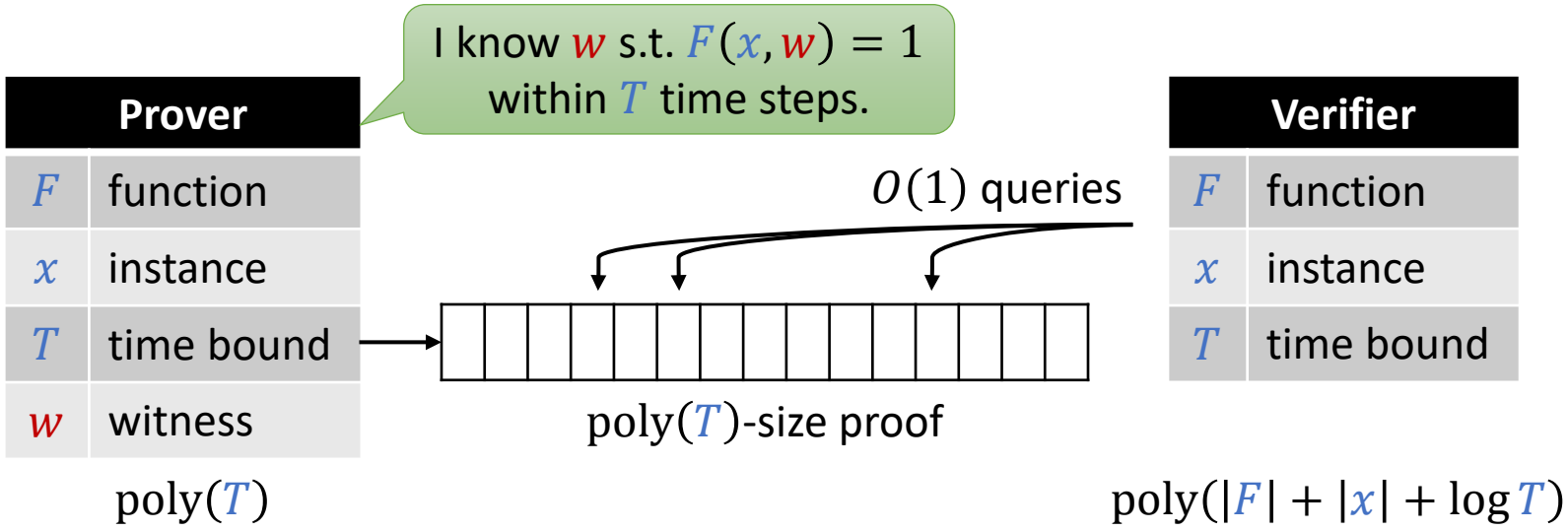


Models

- Qualitative features:
  - **IP**: primarily sub-routines (e.g. sumcheck) to other probabilistic proofs.
  - **PCP**: pedagogically useful but mostly inefficient (e.g. with point queries).
  - **MIP** (& **MIOP**): attractive features (e.g. space efficiency) but hard to use.
  - **IOP**: underlie most efficient SNARKs.

Probabilistically Checkable Proofs

- The verifier is **probabilistic** and has **oracle access** to **1** prover message.



Note: PCP ≠ Succinct Argument!

It is insecure for the verifier to just ask the prover to answer a few queries.

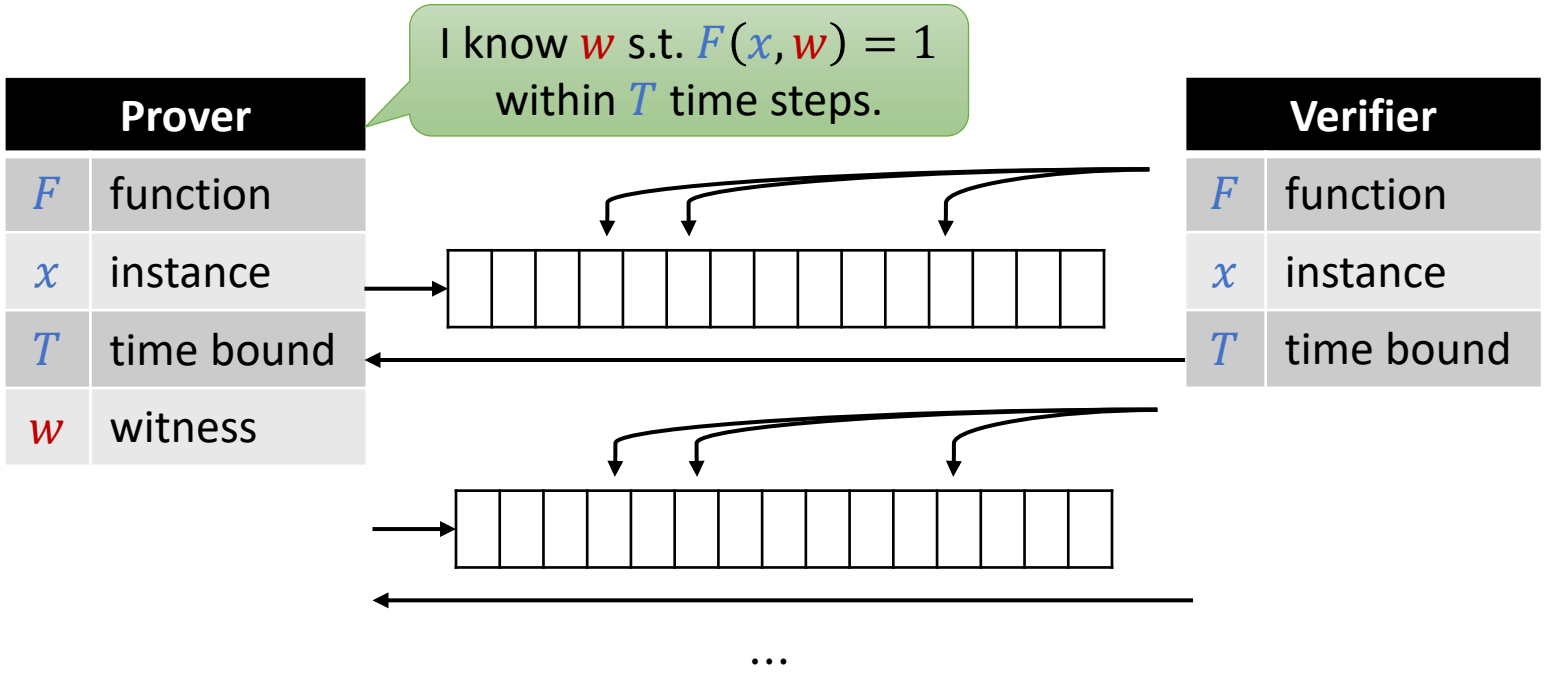
Most PCPs are Inefficient

- **1990s - 2010**: PCPs are galactic (asymptotically efficient but concretely useless).
- **2010 - 2013**: galactic → expensive.
  - PCP-based SNARGs where argument size is 10s of MBs & non-trivial for large (but not galactic) values of T.
- **2013 - now**: slow PCP improvements.

Notable Exception: PCPs with linear queries are efficient.

Interactive Oracle Proofs

- The verifier can simultaneously leverage **randomness**, ***interaction***, and **oracle access**.



Constructions of IOPs

- Flurry of IOP research in the past few years:
  - quasilinear-time ZK [BCGV16][BCFGRS17].
  - linear-size proof length [BCGRS17][RR20].
  - linear-time prover [BCGGHJ17][BCG20][BCL20].
  - linear-time proximity proofs [BBGR16][BBHR18][BKS18][BGKS20][BCIKS20][BN20].
  - efficient implementations [BBC+16][BBHR19][BCRSVW19][COS20].
- Many new techniques:
  - Interactive proof composition.
  - Univariate sumcheck.
  - Out-of-domain sampling.
  - Algebraic linking.

IOPs offer much improved efficiency (asymptotically & concretely).

Realizing Proof Models: Cryptography

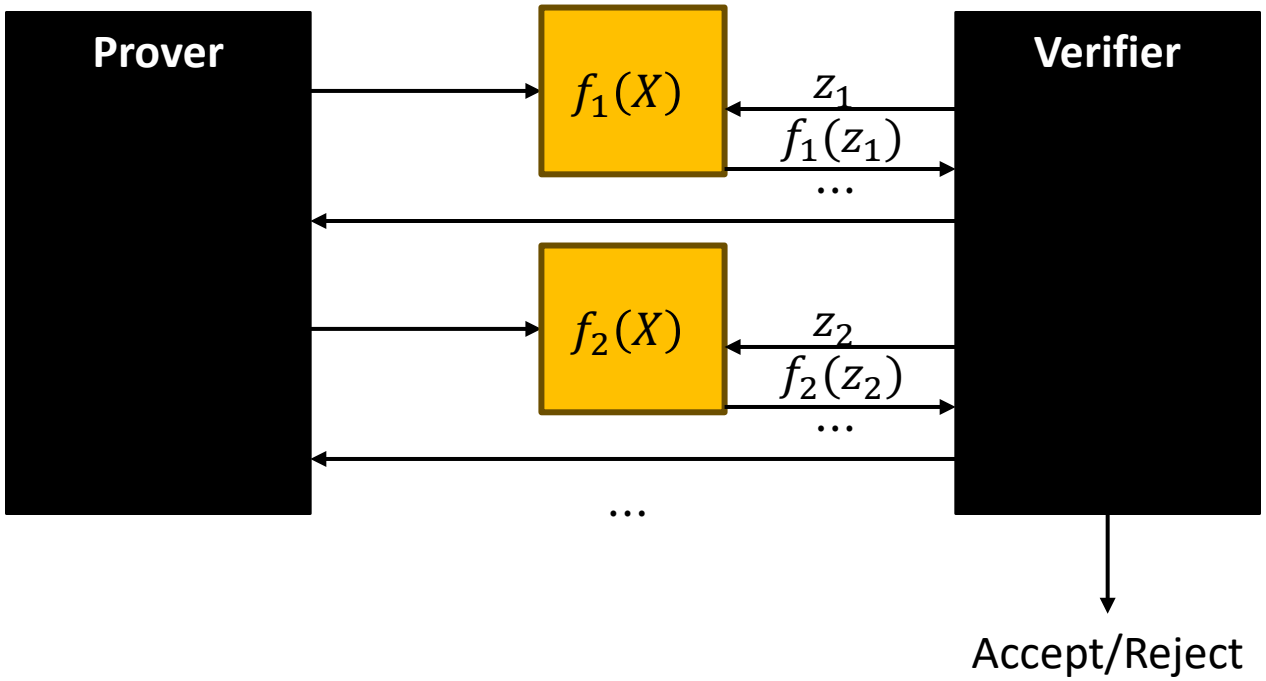
- Examples of SNARK recipes:

Probabilistic Proof	Cryptography	SNARK
linear PCP (and 2-message linear IP)	**linear encoding**	[G10][L11][BCIOP13] [GGPR13][PGHR13] [G16][GM17]…
PCP and IOP	**vector commitment**	Ligero, Aurora, Fractal, SCI, STARK, ...
Polynomial PCP & IOP	**polynomial commitment**	Sonic, Marlin, Plonk, Spartan Supersonic-RSA, Hyrax, vSQL, vRAM, Libra, …

↓ type of computation (e.g., circuit vs machine)

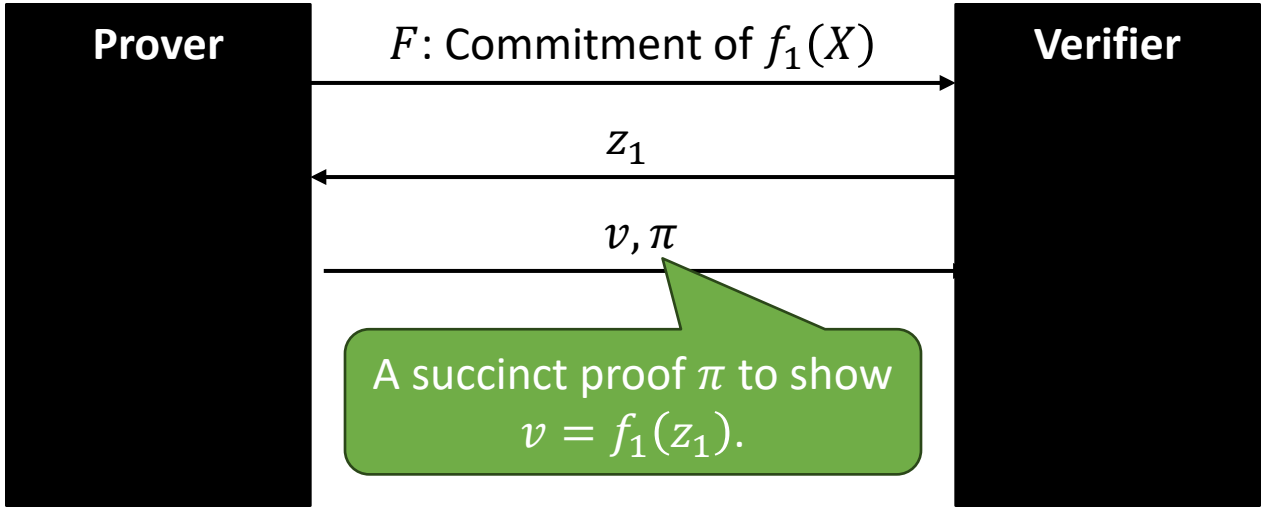
↓
- cryptographic costs (in prover and in verifier)
- pre-quantum or post-quantum
- setup (public or private, specific or universal)

Polynomial IOP



It can also support multi-variant polynomials.

PCS Relation



$$\mathfrak{R} = \left\{ \begin{matrix} \text{witness: } f_1(X) \\ \text{public input: } F, z_1, v \end{matrix} \middle| \begin{matrix} F = Com(f_1) \\ v = f_1(z_1) \end{matrix} \right\}.$$

Can you write the relation for multivariant polynomial PCS?

Thanks!

Q&A