KZH-Fold: Accountable Voting from Sublinear Accumulation

George Kadianakis - Aranxta Zapico - Hossein Hafezi - Benedikt Bunz

ethereum foundation

Background

Argument of Knowledge

Instance/witness (x, w)

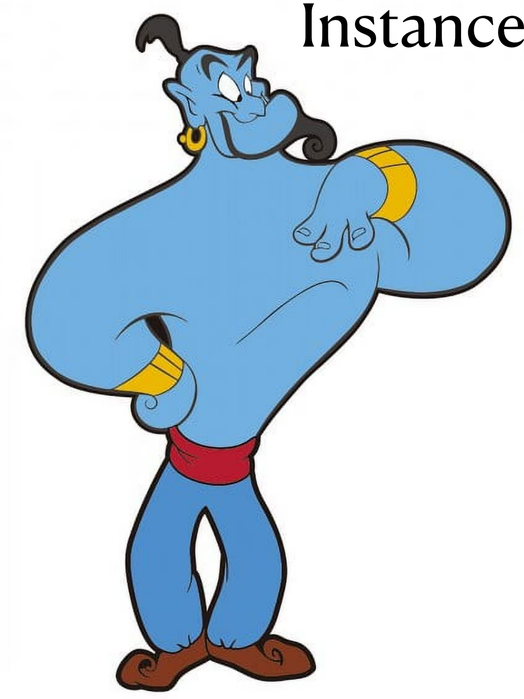
The prover

Instance x

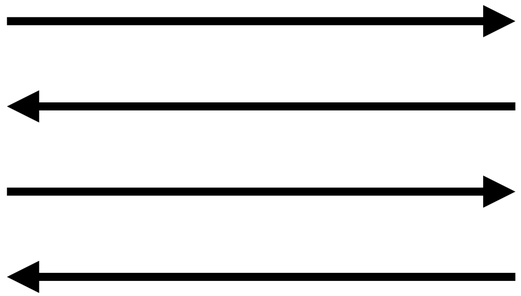
The verifier

Argument of Knowledge

Instance/witness (x, w)



The prover



Instance x

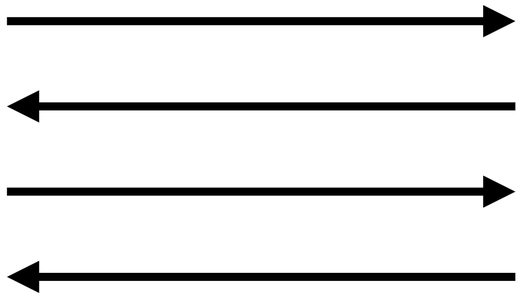
Genie knows witness w

The verifier

Argument of Knowledge

Instance/witness (x, w)

The prover



Instance x

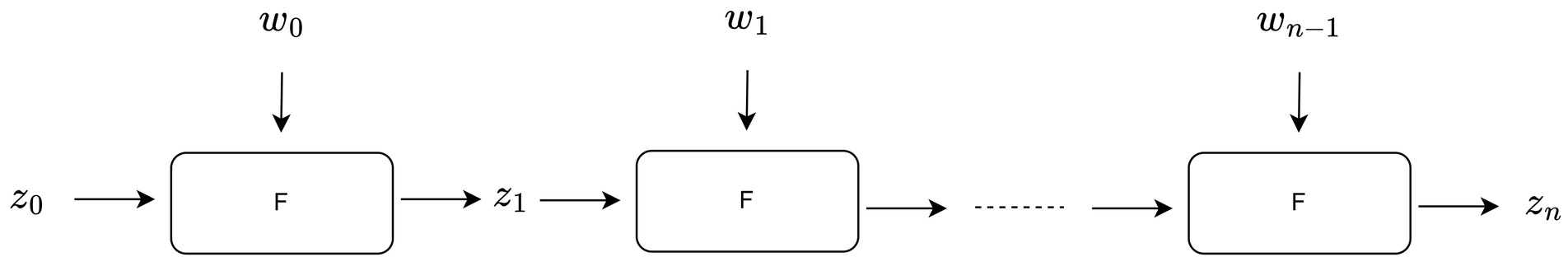
The verifier

Genie knows witness w

Non-interactive ==> NARK

Incremental Verifiable Computation (IVC)

- A class of arguments of knowledge for iterative computation of a function F



- Proof-Carrying Data (PCD): A generalization of IVC to directed acyclic graphs

Distributive proving

IVC Construction from Accumulation Schemes [Halo]

- **State-of-the-art IVC/PCD scheme ==> Built from accumulation schemes**
- Accumulation Scheme [BCMS21;KST21;KP22; BC23]:

Let $\mathcal{L}_\pi$ and $\mathcal{L}_{acc}$ be two NP languages.

An accumulation:

$$acc \in \mathcal{L}_{acc}, \pi \in \mathcal{L}_\pi \overset{\text{accumulate}}{\Longrightarrow} acc' \in \mathcal{L}_{acc}$$

satisfies *completeness* and *knowledge soundness*.

Accumulation Scheme

Completeness: acc and π are satisfied $\iff$ acc' is satisfied.

- Decider $=$ Satisfiability function of $\mathcal{L}_{acc}$

Knowledge soundness: witness for acc' $\implies$ witness for acc and π.

Accumulation Scheme

Completeness: acc and π are satisfied $\iff$ acc' is satisfied.

- Decider $=$ Satisfiability function of $\mathcal{L}_{acc}$

Knowledge soundness: witness for acc' $\implies$ witness for acc and π.

Sublinearity:

- $\text{acc} \in \mathcal{L}_{acc}, \pi \in \mathcal{L}_{\pi} \implies |acc| \in o(|\pi|)$
- Decider time $<$ Verification of π

BLCMS Compiler: NARK + NARK Verifier Accumulation ==> IVC

BCLMS Compiler: NARK + NARK Verifier Accumulation $\implies$ IVC

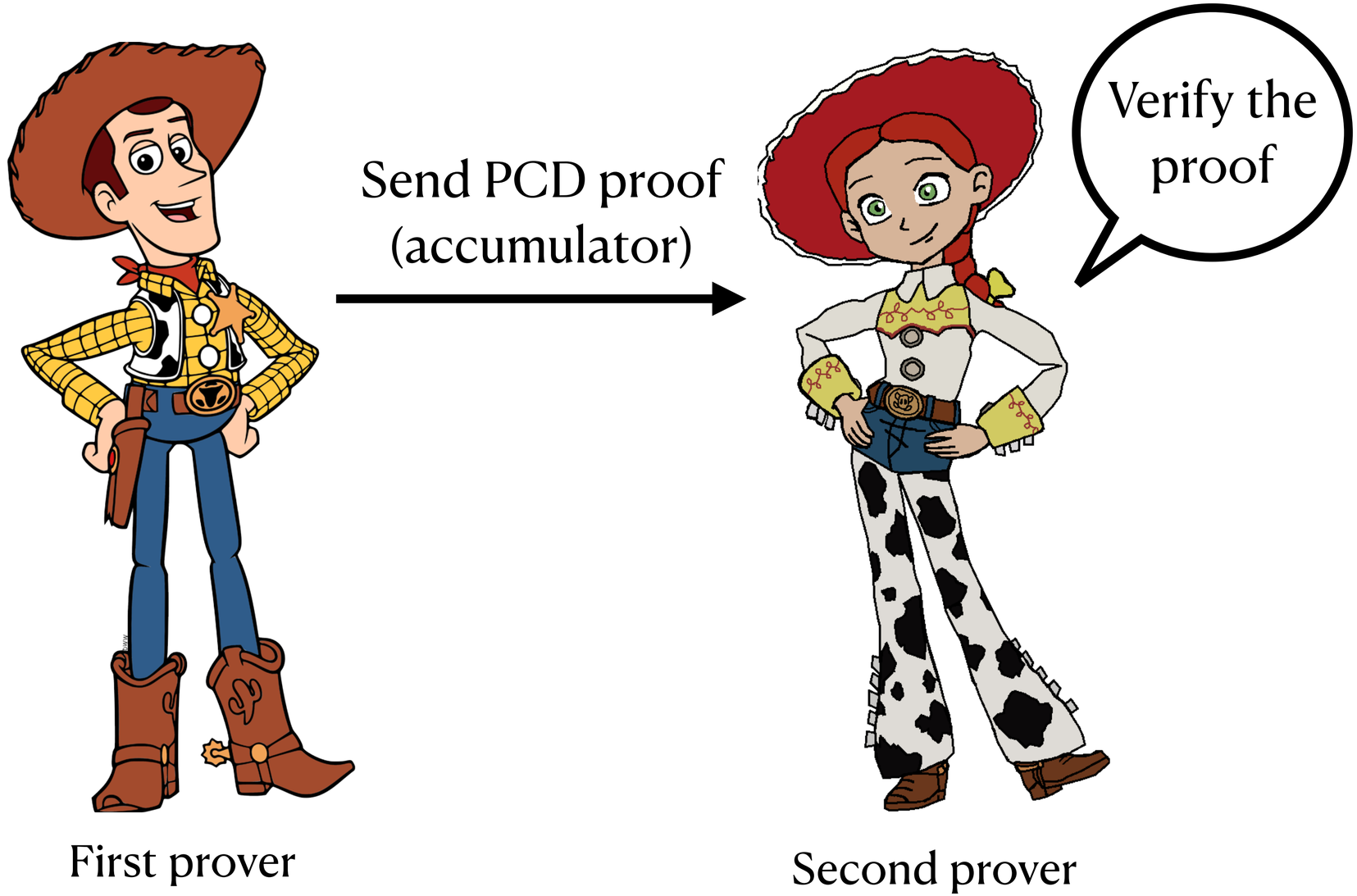
- IVC proof = $|\text{acc}|$
- IVC verifier = $\text{Decider}_{\text{acc}}$
- IVC prover = $\text{Prover}_{\text{NARK}}$ (for augmented circuit) + $\text{Prover}_{\text{acc}}$

Sublinear accumulator $\implies$ IVC/PCD with sublinear proofs/decider

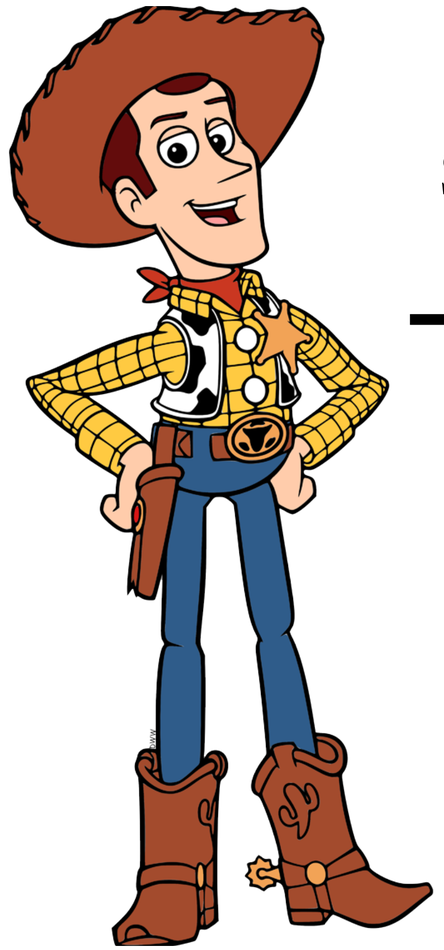
Why Sublinearity Matters?

- Consider a function (circuit) F to be **1M** constraints, using Nova [KST22]:
  - IVC proof size: **80MB**
  - IVC verifier time: **4s**
- MicroNova/Nova solution: Run a zkSNARK at the last step of IVC ==> reduces proof size **1KB**
  - The resulting proof is no longer incremental!
  - **24x** prover overhead to run the zkSNARK compared to a single IVC step

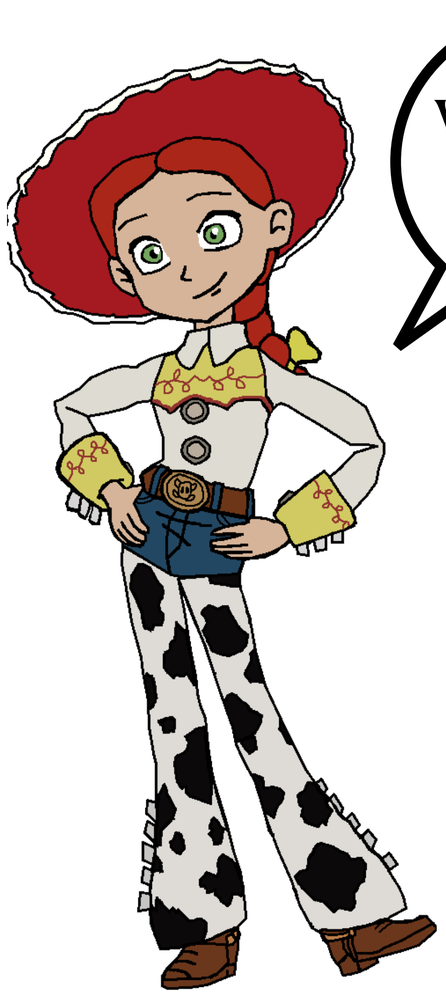
Distributed Proving with PCD



Motivation: Distributed Proving with PCD



Send PCD proof
(accumulator)



Verify the proof

First prover

Second prover

For linear-sized accumulation scheme, both communication and verifier time are high!

Our Contributions

- **KZH:** PCS with Sublinear Opening
- **KZH-fold:** Sublinear Accumulator for KZH
    - IVC/PCD with sublinear proof and verifier.
    - Signature aggregation for Ethereum via PCD.
- New Approach to non-uniform PCD ==> we do not get into it in the talk.

KZH: Polynomial Commitment With Optimal Opening

For a multilinear polynomial $f(\vec{X})$ with on boolean hypercube of size ℓ:

- commitment time $= O(\ell)$ group operations.
- Proof size $= \sqrt{\ell}\ \mathbb{G}_1 + \sqrt{\ell}\ \mathbb{F}$
- Opening time $= \ell\ \mathbb{F}$ [1]
- Verifier time $= \sqrt{\ell}$ pairing $+$ MSM$(2\sqrt{\ell}) + \sqrt{\ell}\ \mathbb{F}$.

We extend KZH matrix construction to tensors of higher degree $\implies$ Lower verifier cost at the cost of higher opening.

- Commitment cost: $O(k \cdot n)$
- Opening cost: $O(n^{1/2})$ via pre-processing
- Verifier cost: $O(k \cdot n^{1/k})$

KZH-fold: Accumulating KZH Verifier

The verifier function for KZH$=$ degree 2 scalar multiplications $+$ degree 1 pairing check $\overset{\text{Protostar compiler}}{\Longrightarrow}$

- Accumulator size. $O(\ell^{\frac{1}{2}})$
- Decider complexity. $O(\ell^{\frac{1}{2}})$
- Prover complexity. $O(\ell)$
- Verifier complexity. $3-4$ $\mathbb{G}_1$ scalar multiplications $+$ $O(1)$ $\mathbb{F}$

KZH-k fold:

- $O(k \cdot n^{1/k})$ decider and accumulator size
- $k+1$ scalar multiplication in recursive circuit.

See Some Numbers

Scheme	Prover	Verifier	$\lvert\mathsf{Acc}\rvert$	# Constraints
Spartan+KZH-fold	16.5 s	135 ms	37 KB	$2^{21} \approx 2097k$
Nova	4.8 s	5.6 s	80.8 MB	1185k

Table: Spartan+KZH-Fold vs Nova for Circuit With 2000 Poseidon Hashes

Thank you!