Lite-PoT

Practical Powers-of-Tau Setup Ceremony

Lucien K. L. Ng[12], Pedro Moreno-Sanchez[234], Mohsen Minaei[2],

Panagiotis Chatzigiannis[2], Adithya Bhat[2], Duc V. Le[2]

[1]Georgia Institute of Technology [2] VISA Research

[3] IMDEA Software Institute [4] Max Planck Institute for Security and Privacy

Powers of Tau (PoT)

$$\mathrm{pp} = \left(\tau I_{\mathbb{G}_1}, \tau^2 I_{\mathbb{G}_1}, \ldots, \tau^n I_{\mathbb{G}_1};\ \tau I_{\mathbb{G}_2}, \ldots, \tau^k I_{\mathbb{G}_2}\right) \in \mathbb{G}_1^n \times \mathbb{G}_2^k$$

Applications of PoT

- PoT is essential for KZG-based zk-SNARK and many blockchain apps
  - Data-Sharding (Proto-Danksharding for Ethereum)
  - Privacy-Preserving Cryptocurrencies/Transactions (ZCash, Aleo, etc.)
  - Layer-2 Rollups (zkSync, zkRollup)
  - Rate-Limiting Nullifiers for Spam Protection
- Other cryptographic primitives
  - Accumulators, Verkle Trees, Multi-Writer encrypted Databases, …

KZG: Kate-Zaverucha-Goldberg (KZG) polynomial commitment scheme
zk-SNARK: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

Trusted Setup for PoT

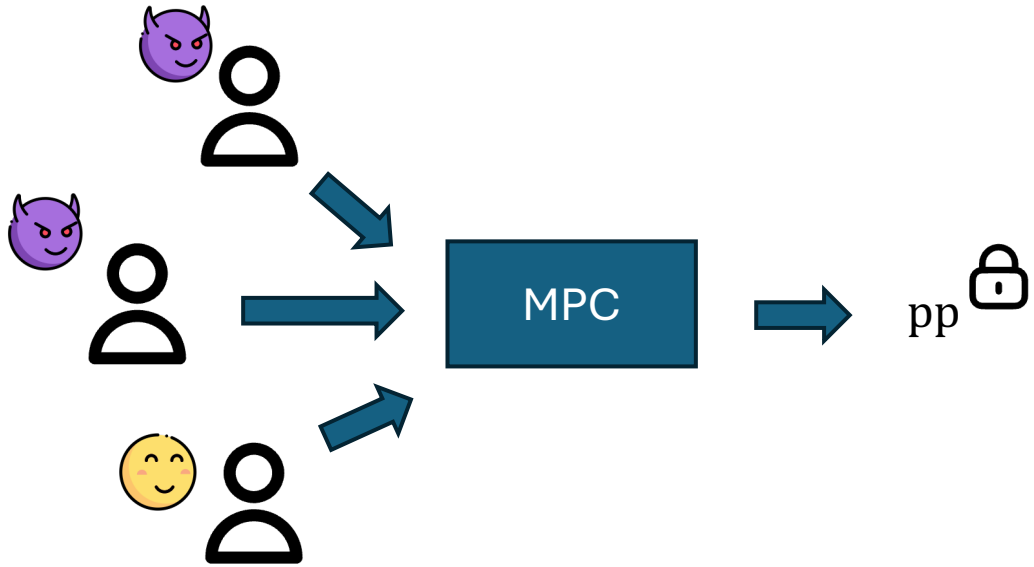
- PoT is a public parameter pp produced during setup phase
  - We will not talk about how PoT enables the mentioned primitives/apps

- 😭 Important Fact: PoT requires a trusted setup
  - The trusted setup generates pp and a "toxic waste" τ
  - Trusted setup means users trust that τ is *discarded* after the setup
  - Anyone with τ can forge proofs and thus *destroy* the security

$$\mathrm{pp} = \left(\tau I_{\mathbb{G}_1}, \tau^2 I_{\mathbb{G}_1}, \ldots, \tau^n I_{\mathbb{G}_1};\ \tau I_{\mathbb{G}_2}, \ldots, \tau^k I_{\mathbb{G}_2}\right) \in \mathbb{G}_1^n \times \mathbb{G}_2^k$$

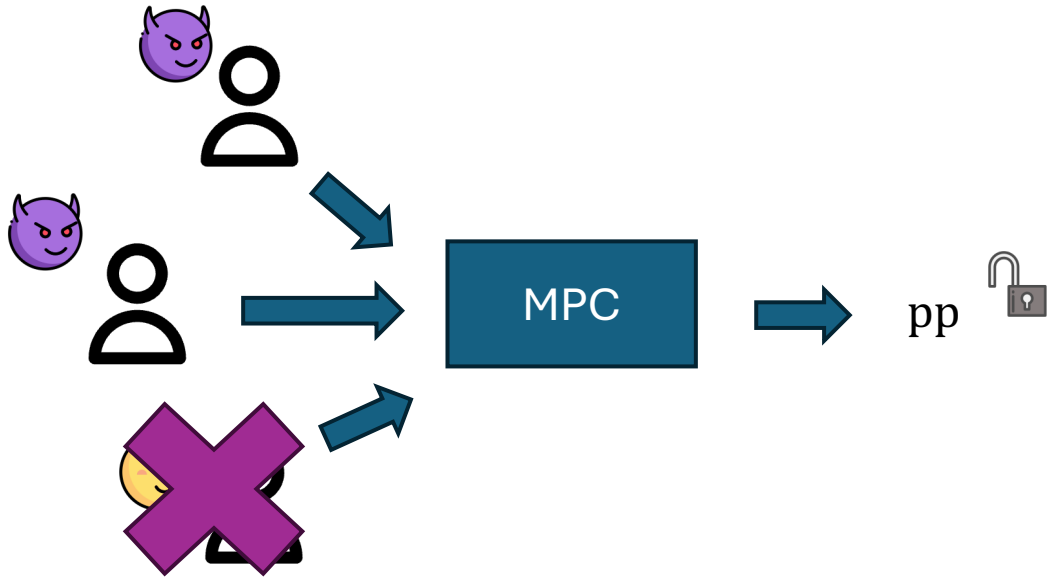
Centralized Setup

- Naïve solution: let a group of parties run MPC to produce pp
  - MPC: Multi-party Computation
  - Anyone can contribute its randomness to pp
  - As long as one user is honest, τ is secret
  - More contributors are better (the more-the-merrier)



Censorship in Centralized Setup

- 😭 Major Issue: Censorship
  - The parties can exclude anyone they do not like
  - By censorship, an adversary can control all the randomness (thus τ) in pp
  - Users are less confident in the security of PoT-based applications



Decentralized Setup Ceremony

- The ceremony takes place on an Ethereum's smart contract
  - Anyone can participant by submitting its contribution to the contract

Smart Contract is Expensive

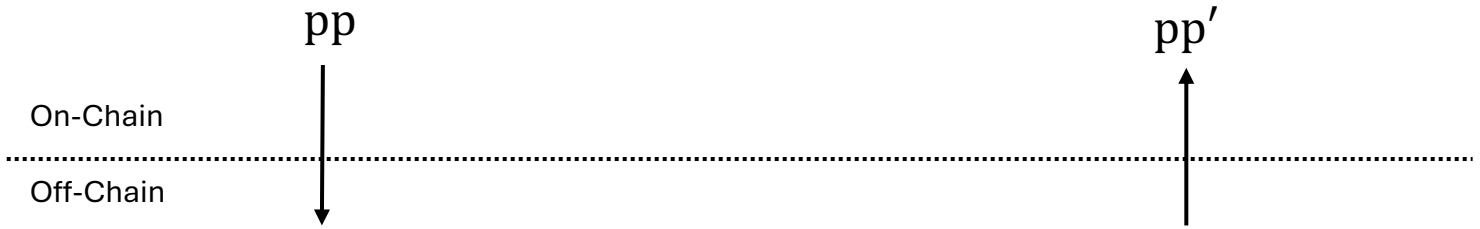
- Major Issue: The contract's gas cost is expensive
  - It discourages randomness contributors
  - It limits the size of pp
    - Useful applications usually need larger pps

Our Goals

Can we significantly reduce the monetary cost for participating in a decentralized PoT setup ceremony?

- More contributors can join ⇒ Users have more confident in PoT

How Prior Art [ACNS'24] Works



1. Parses $pp \rightarrow (P_1, P_2, \ldots; Q_1, \ldots)$
2. Samples $r \leftarrow_{\$} \mathbb{Z}_p$
3. $\text{pp}' \leftarrow (r \cdot P_1, r^2 P_2, \ldots; r \cdot Q_1, \ldots)$

Valeria Nikolaenko, Sam Ragsdale, Joseph Bonneau, Dan Boneh. Powers-of-Tau to the People: Decentralizing Setup Ceremonies, ACNS '24

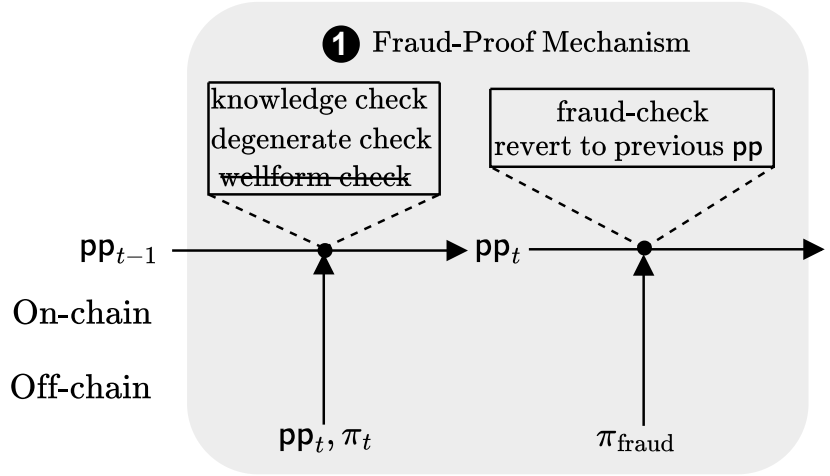
Why is Prior Art [ACNS'24] Gas-Inefficient

- Anyone can update pp to pp' in on-chain if it can pass
  - Well-formedness Check: pp' is well-formed
    - pp' are really powers-of-tau that $\left(\tau I_{\mathbb{G}_1}, \ldots, \tau^n I_{\mathbb{G}_1};\ \tau I_{\mathbb{G}_2}, \ldots, \tau^k I_{\mathbb{G}_2}\right)$
    - 🙁 Expensive $O(n\ +\ k)$ Elliptic Curve (EC) Operations
  - Knowledge Check: it knows r
    - Without it, an adv can reset pp by sampling τ^* and setting $\mathrm{pp}' = \left(\tau^* I_{\mathbb{G}_1}, \ldots;\ \tau^* I_{\mathbb{G}_2}, \ldots\right)$
    - 🙁 Only one can update pp on-chain at a time
  - Non-degeneration Check: τ' is not zero (trivial)

Valeria Nikolaenko, Sam Ragsdale, Joseph Bonneau, Dan Boneh. Powers-of-Tau to the People: Decentralizing Setup Ceremonies, ACNS '24

1st Idea: Fraud-Proof Mechanism

- Optimistic Update
  - A contributor only upload new pp.
  - The smart contract does not run the (expensive) well-formedness check
- Fraud Proof
  - A challenger can revert the contract to previous valid pp
  - The cost is only a membership proof + an algebraic check



Concrete Cost Estimation

- Prior gas cost: $O(n)$ storage + $O(n)$ ECMULT
- Ours' gas cost:
  - Optimistic Update: $O(1)$ storage + $O(n)$ hashing
  - Fraud Proof: $O(\log n)$ hashing + $O(1)$ ECMULT
- We can support 2^{15}-degree pp (vs. 2^{11} in the prior art)
  - 😄 We can support more complex PoT-based applications

Operation	Gas Cost
Hashing 32-bytes words	36
Reading 32-btyes contract input	512
ECMULT	6,000
Storing a 32-bytes word	20,000

n is the degree of pp

n	2^{11}	2^{13}	2^{15}
Total Cost of Prior Art	23.1M **(75 USD)**	127.3M **(415 USD)**	851M **(2774 USD)**
Our Optimistic Update Cost	3.9M **(13 USD)**	7.9M **(26 USD)**	25.4M **(83 USD)**
Our Fraud Proof Verification Cost	325,516 **(1 USD)**	332,051 **(1 USD)**	338,647 **(1 USD)**

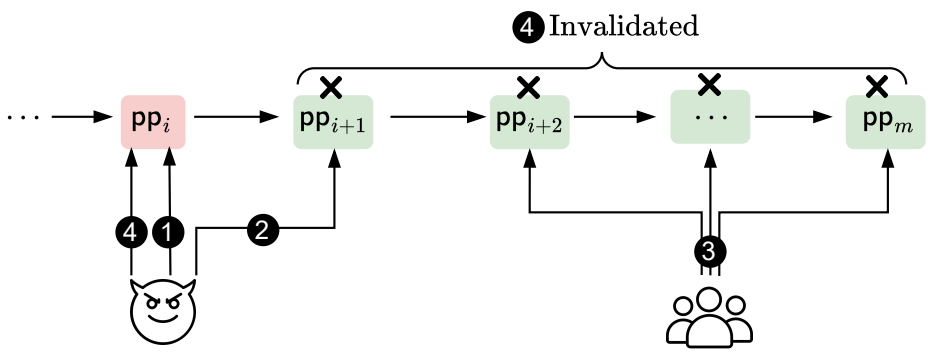
* The USD values were calculated based on the Ethereum price of 1,630 USD and gas price of 2 gwei on April 14th, 2025.

33x

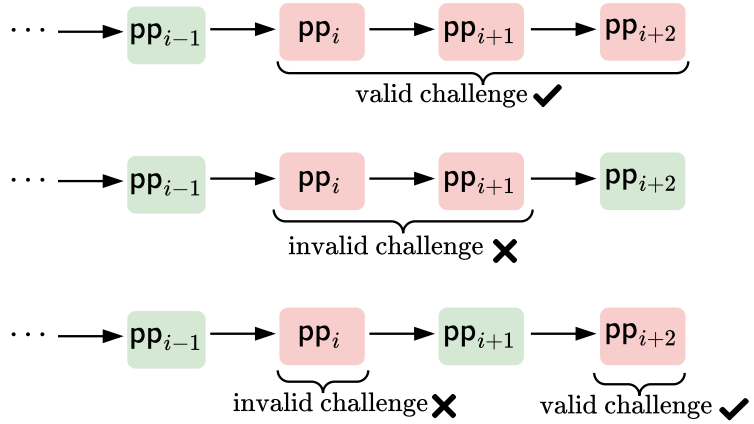
Fraud proof is cheap

Optimization:
Suffix-only Challenge Mechanism

- Goal: To reduce *off-chain* verification costs



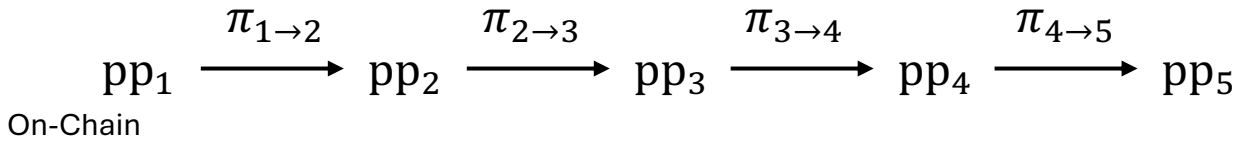
An attack in Naïve Design
when contributors don't check all historic pp



Our mechanism to protect contribution
even if the contributor only check the last pp

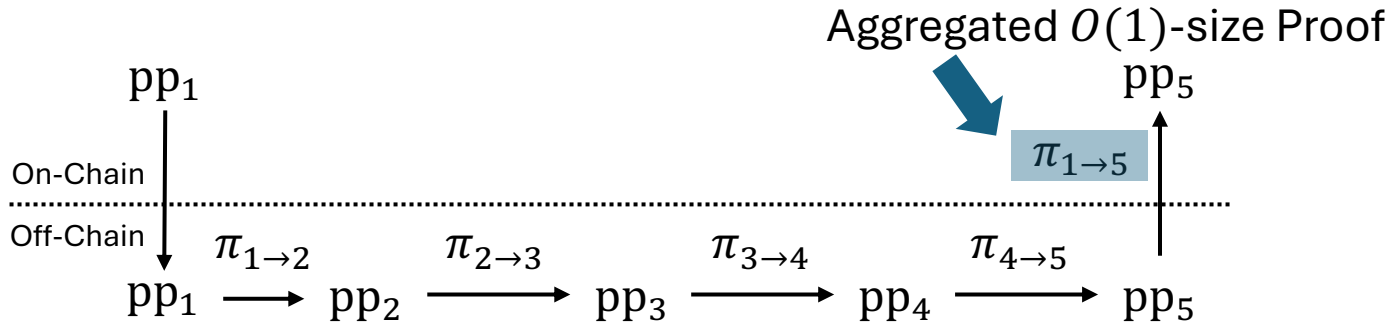
Idea 2: Batching Contributions

Prior Art



- $O(m)$ Gas for m contributions
  - For knowledge check: each contributor knows r
  - Fraud-proof idea cannot save this part

Ours

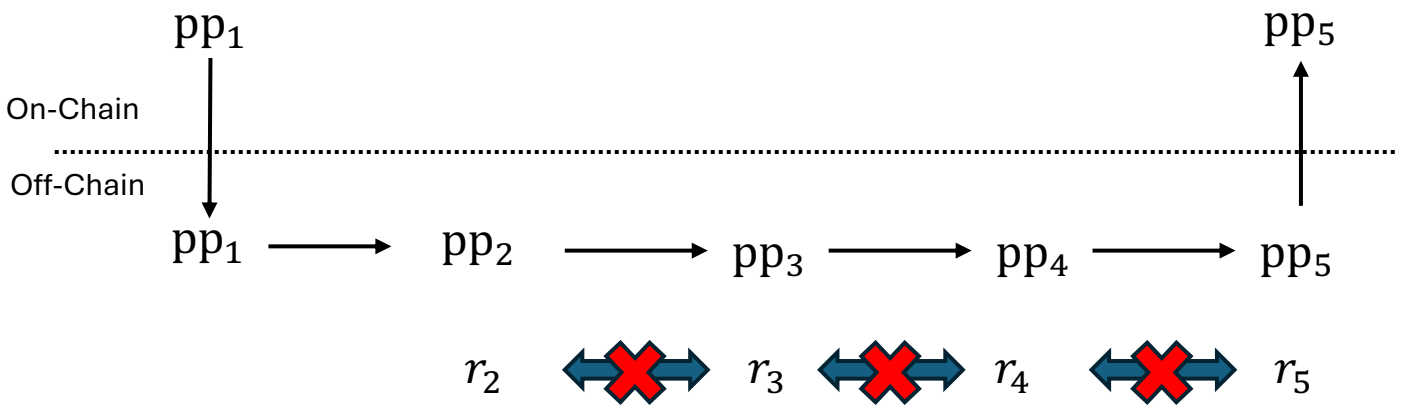


How Large is #Contributions (m)?

- Ethereum's (centralized) KZG ceremony has m>140,000 contributors
  - m might get even larger in the future
- In the prior art, each knowledge check takes >10,000 gas
  - So >1.4B gas in total, which exceeds the gas limit (30M) by >46x

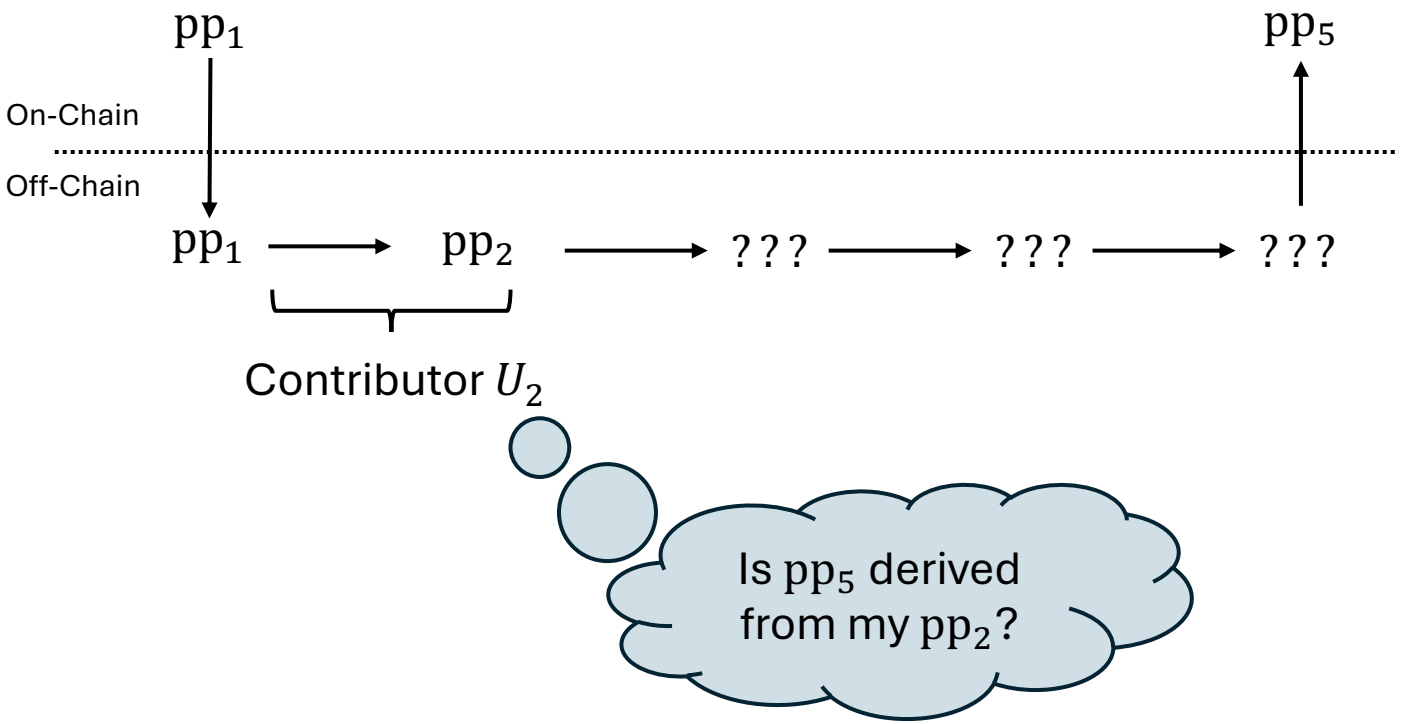
Verifying our aggregated proof costs only constant gas!

Challenge 1: Oblivious Aggregate

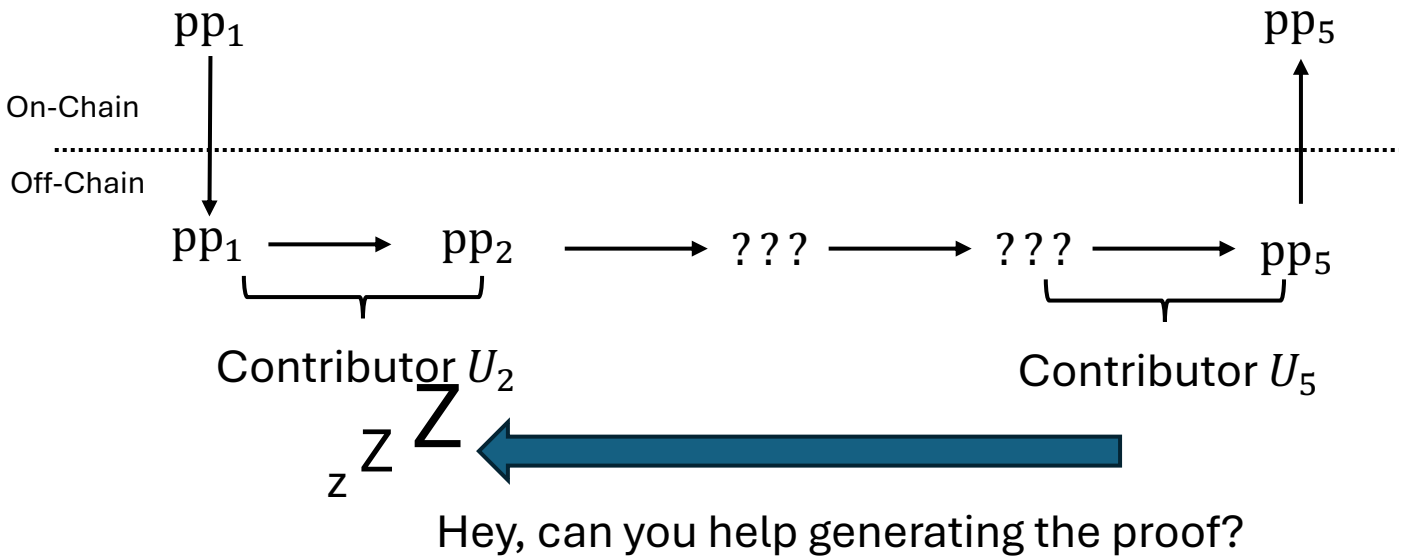


- Check #1: it knows $r = r_2 \cdot r_3 \cdot r_4 \cdot r_5$
- But no one knows $r_2 \cdot r_3 \cdot r_4 \cdot r_5$

Challenge 2: Inclusion of Contribution



Challenge 3: Non-Interactiveness



Closing Remark of Lite-PoT

- We proposed a PoT setup ceremony scheme that is
  - Censorship-resistant by using a smart contract
  - Cheap to participant
- Such a ceremony can increase users' confidence in the PoT
- Our Contributions: Fraud Proof Mechanism & Aggregatable Proofs
- Open Question: Can PoT ceremonies make use of data availability solutions, e.g., Blob Data on Ethereum?

- Ad: Another talk @ Applied Crypto #11, Rm201DE, tomorrow 1:30pm
  - Toss: Garbled PIR from Table-Only Stacking