Randomness beacons in theory and practice

Joseph Bonneau

Real World Crypto
March 28, 2025

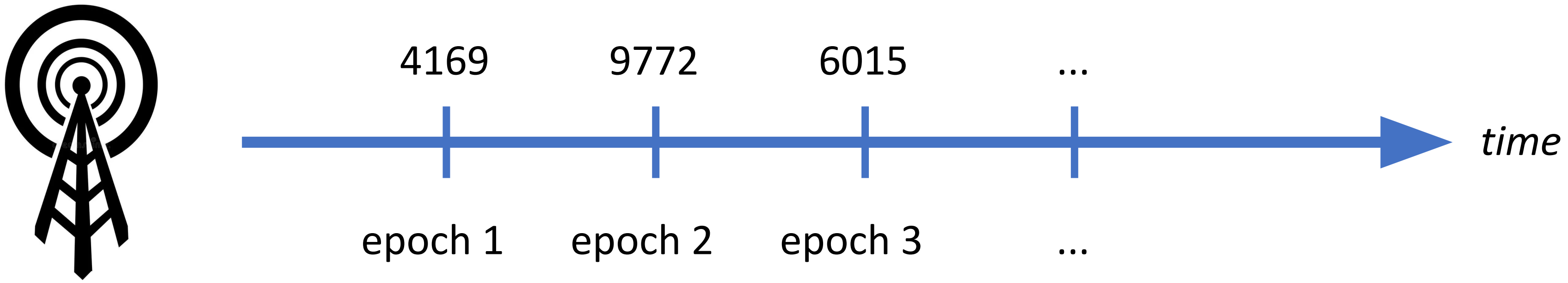
Required disclosures

The views expressed here are those of the individual AH Capital Management, L.L.C. ("a16z") personnel quoted and are not the views of a16z or its affiliates. Certain information contained in here has been obtained from third-party sources, including from portfolio companies of funds managed by a16z. While taken from sources believed to be reliable, a16z has not independently verified such information and makes no representations about the enduring accuracy of the information or its appropriateness for a given situation.

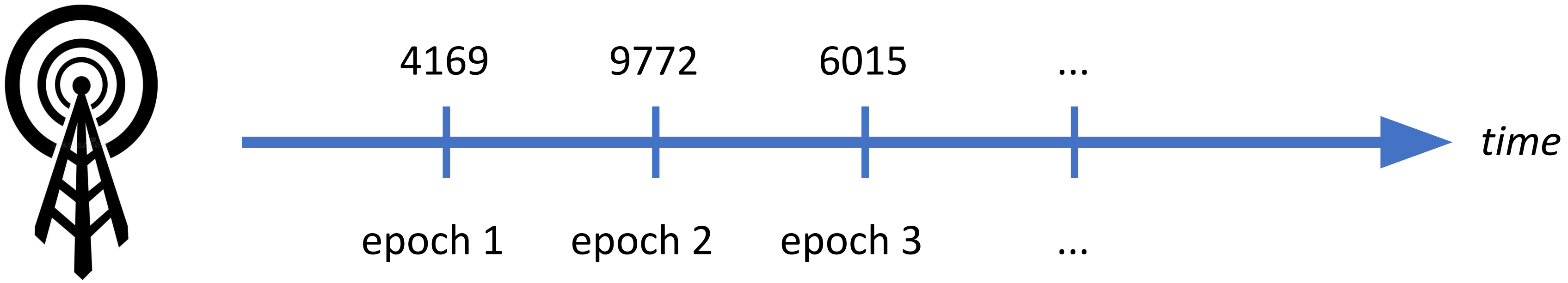
This content is provided for informational purposes only, and should not be relied upon as legal, business, investment, or tax advice. You should consult your own advisers as to those matters. References to any securities, digital assets, tokens, and/or cryptocurrencies are for illustrative purposes only and do not constitute a recommendation to invest in any such instrument nor do such references constitute an offer to provide investment advisory services. Furthermore, this content is not directed at nor intended for use by any investors or prospective investors, and may not under any circumstances be relied upon when making a decision to invest in any fund managed by a16z. (An offering to invest in an a16z fund will be made only by the private placement memorandum, subscription agreement, and other relevant documentation of any such fund and should be read in their entirety.) Any investments or portfolio companies mentioned, referred to, or described are not representative of all investments in vehicles managed by a16z, and there can be no assurance that the investments will be profitable or that other investments made in the future will have similar characteristics or results. A list of investments made by funds managed by Andreessen Horowitz (excluding investments for which the issuer has not provided permission for a16z to disclose publicly as well as unannounced investments in publicly traded digital assets) is available at https://a16z.com/investments/.

Charts and graphs provided within are for informational purposes solely and should not be relied upon when making any investment decision. Past performance is not indicative of future results. The content speaks only as of the date indicated. Any projections, estimates, forecasts, targets, prospects, and/or opinions expressed in these materials are subject to change without notice and may differ or be contrary to opinions expressed by others. Please see https://a16z.com/disclosures for additional important information.

The *randomness beacon* abstraction [Rabin83]



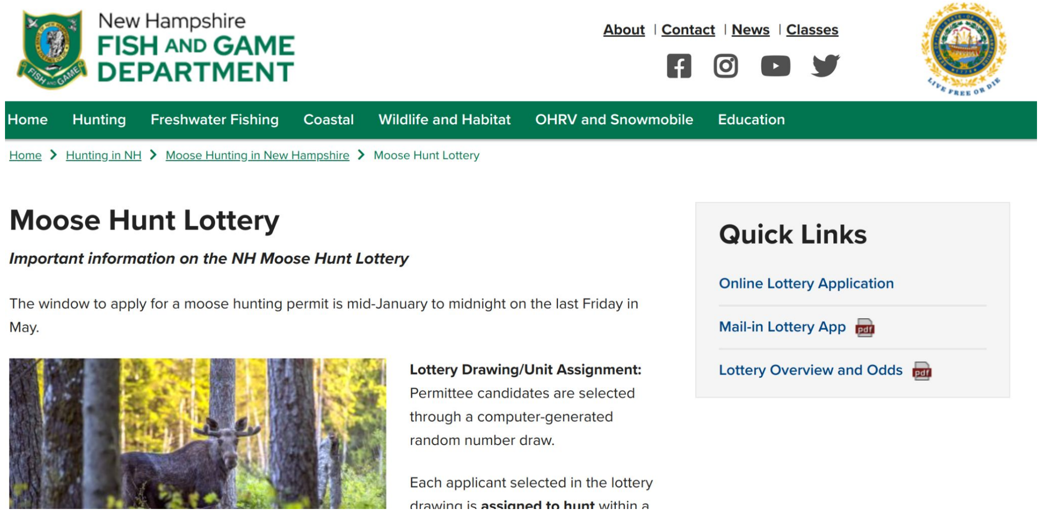
The *randomness beacon* abstraction [Rabin83]



Goals (high level):

- Statistically uniform randomness
- Public consensus on values
- Regular service, high bandwidth
- Attackers can't:
  - Predict
  - Manipulate
  - Block

Beacons can power verifiable lotteries



Many use cases beyond lotteries

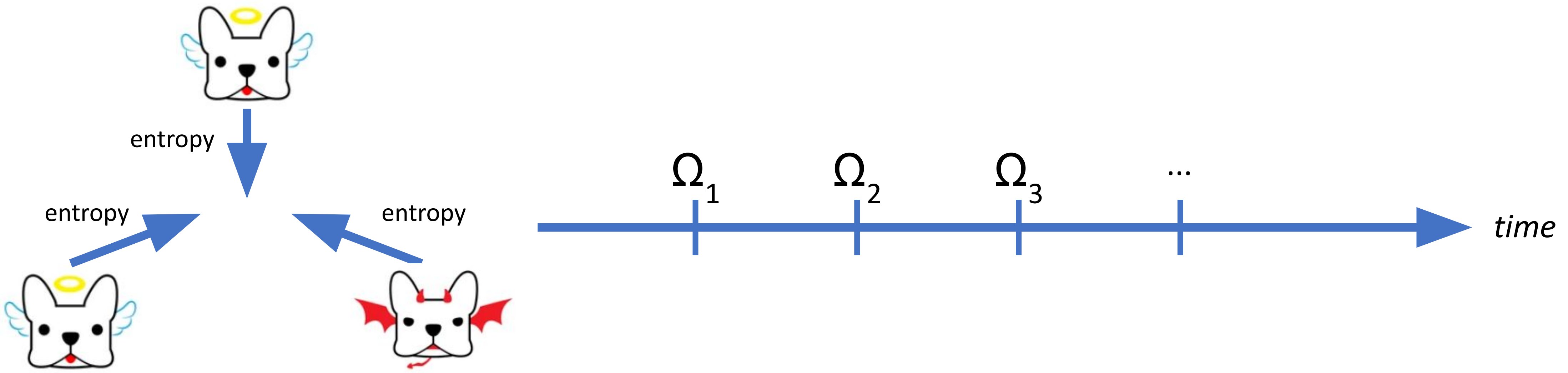
- Games
- Sampling ballots for election audits
- Selecting parameters for cryptographic protocols
- Leader election in BFT consensus & blockchains
- Randomized transaction ordering
- Challenges for non-interactive cryptographic proofs

Goal: Many applications driven by a public randomness beacon

State of the art has barely changed for millenia!

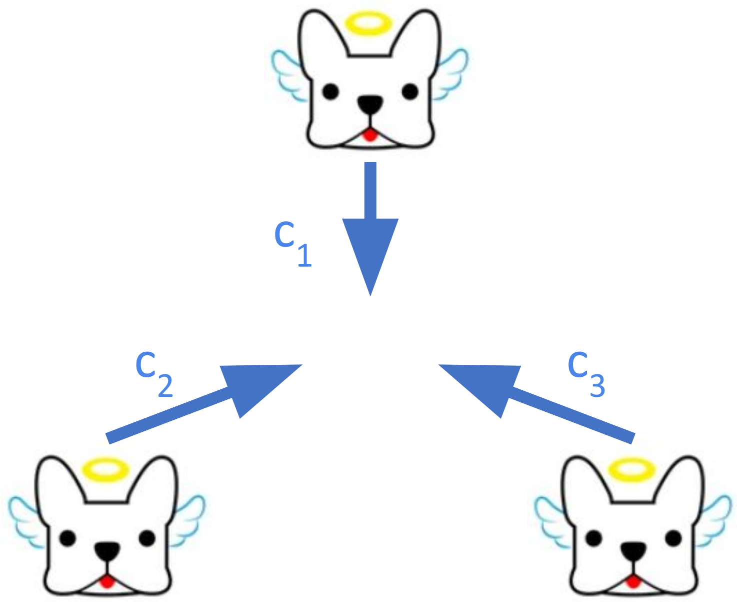
This talk: *distributed* randomness beacons

- Multiparty protocol with n participants, produce output Ω_i in epoch i
- Up to t out of n nodes are controlled by the adversary



Classic: Commit-Reveal

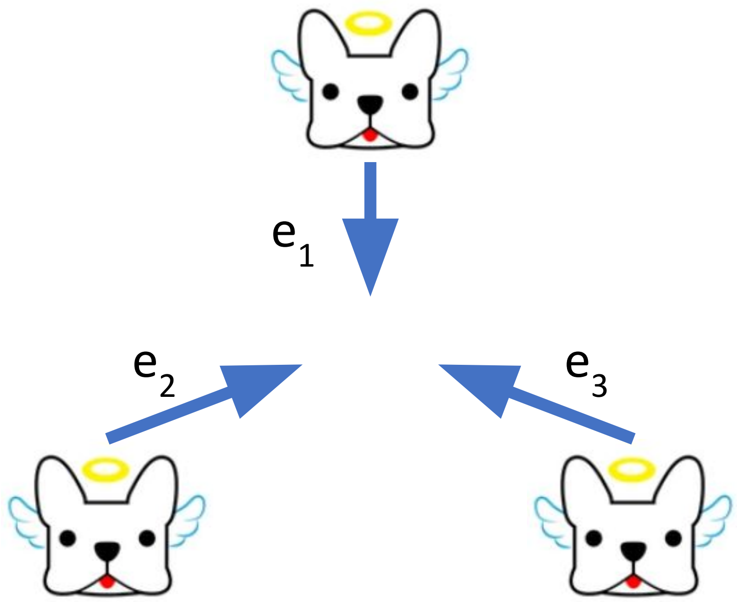
1. Commit
   - Publish a cryptographic commitment $c_i = Commit(e_i)$ to a random value e_i



Classic: Commit-Reveal

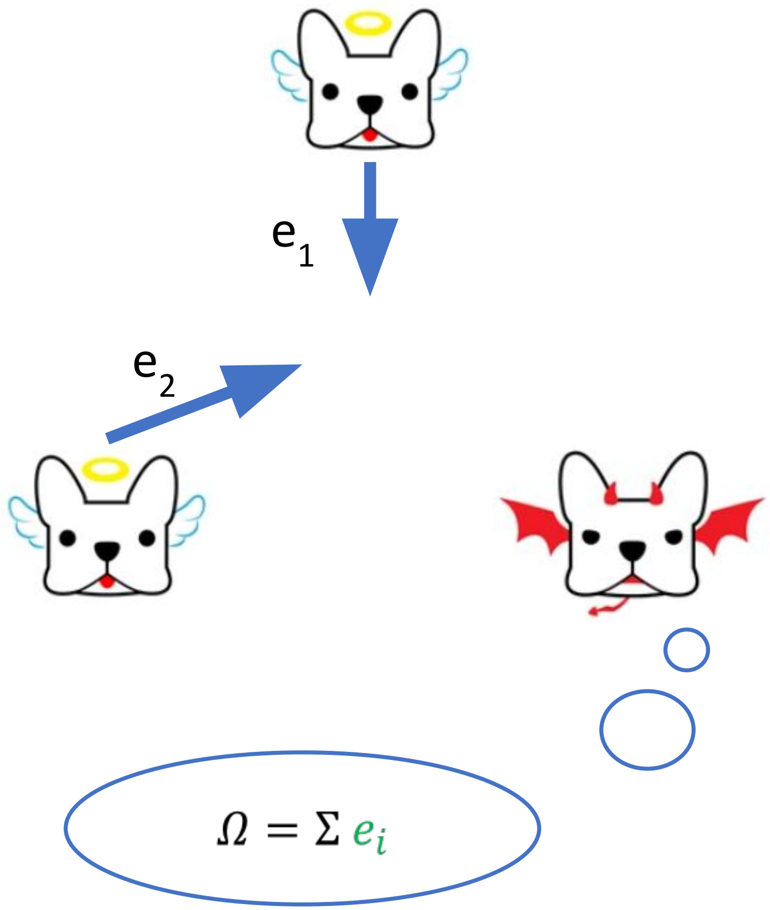
1. Commit
   - Publish a cryptographic commitment $c_i = Commit(e_i)$ to a random value e_i
2. Reveal
   - Participants reveal their e_i values

Beacon output: $\Omega = \Sigma\, e_i$



Classic: Commit-Reveal

1. Commit
   - Publish a cryptographic commitment $c_i = Commit(e_i)$ to a random value e_i
2. Reveal
   - Participants reveal their e_i values

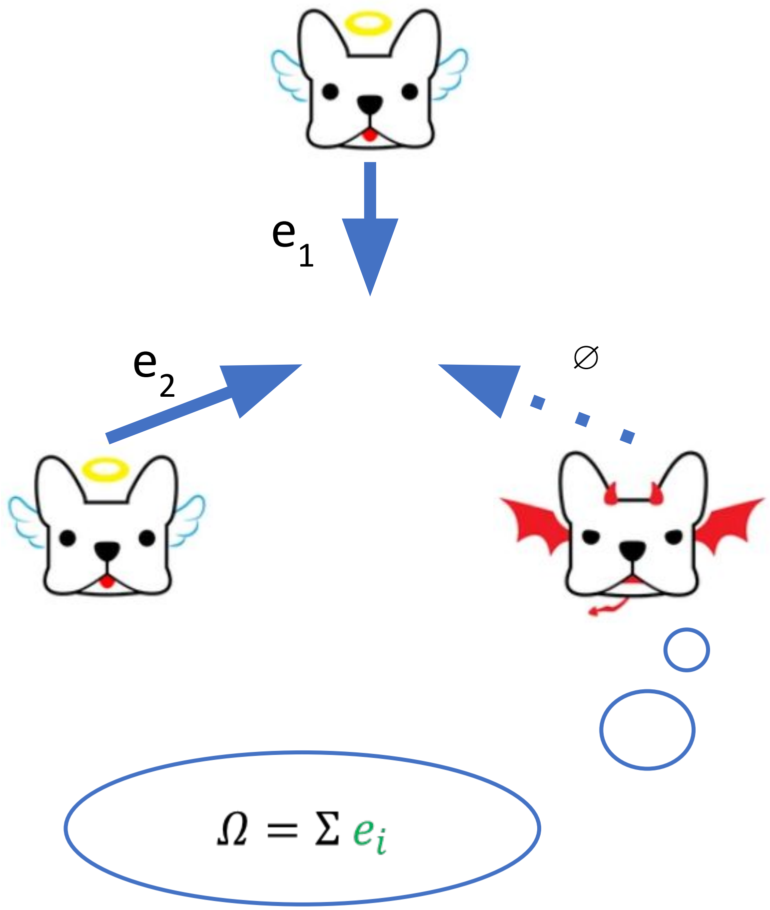


<u>Main problem: last-revealer attack</u>

Classic: Commit-Reveal

1. Commit
   - Publish a cryptographic commitment $c_i = Commit(e_i)$ to a random value e_i
2. Reveal
   - Participants reveal their e_i values

<u>Main problem: last-revealer attack</u>

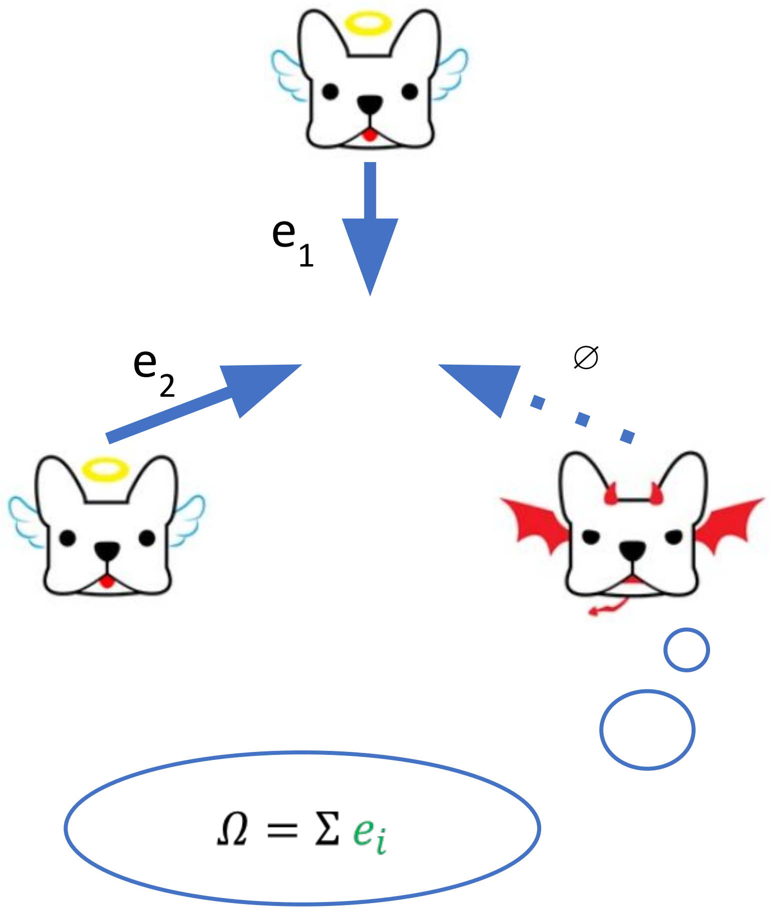


Classic: Commit-Reveal

1. Commit
   - Publish a cryptographic commitment $c_i = Commit(e_i)$ to a random value e_i
2. Reveal
   - Participants reveal their e_i values

Beacon output: $\Omega = \bot$

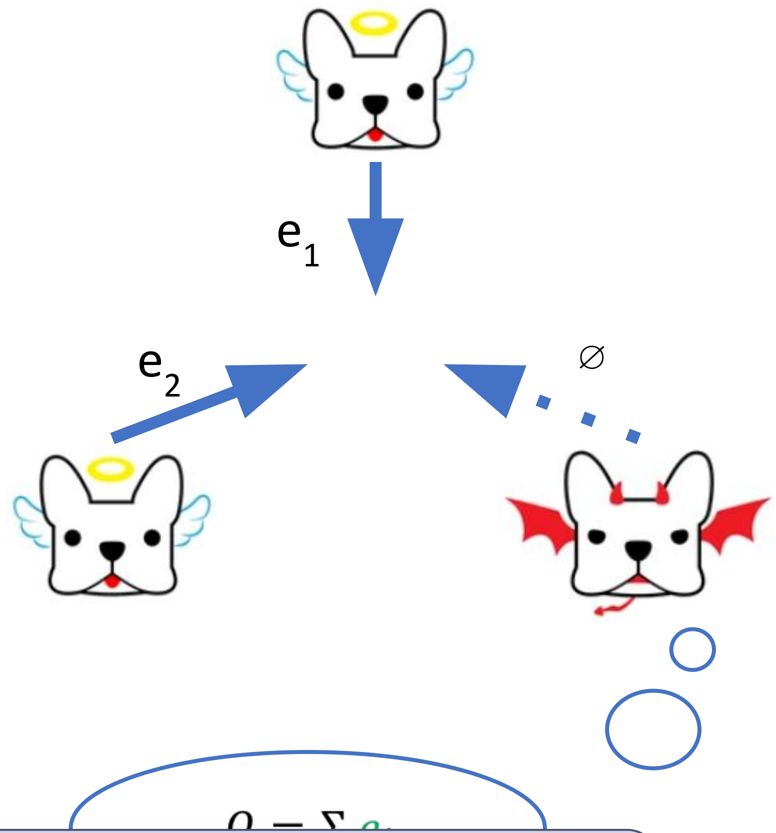
<u>Main problem: last-revealer attack</u>



Classic: Commit-Reveal

1. Commit
   - Publish a cryptographic commitment $c_i = Commit(e_i)$ to a random value e_i
2. Reveal
   - Participants reveal their e_i values

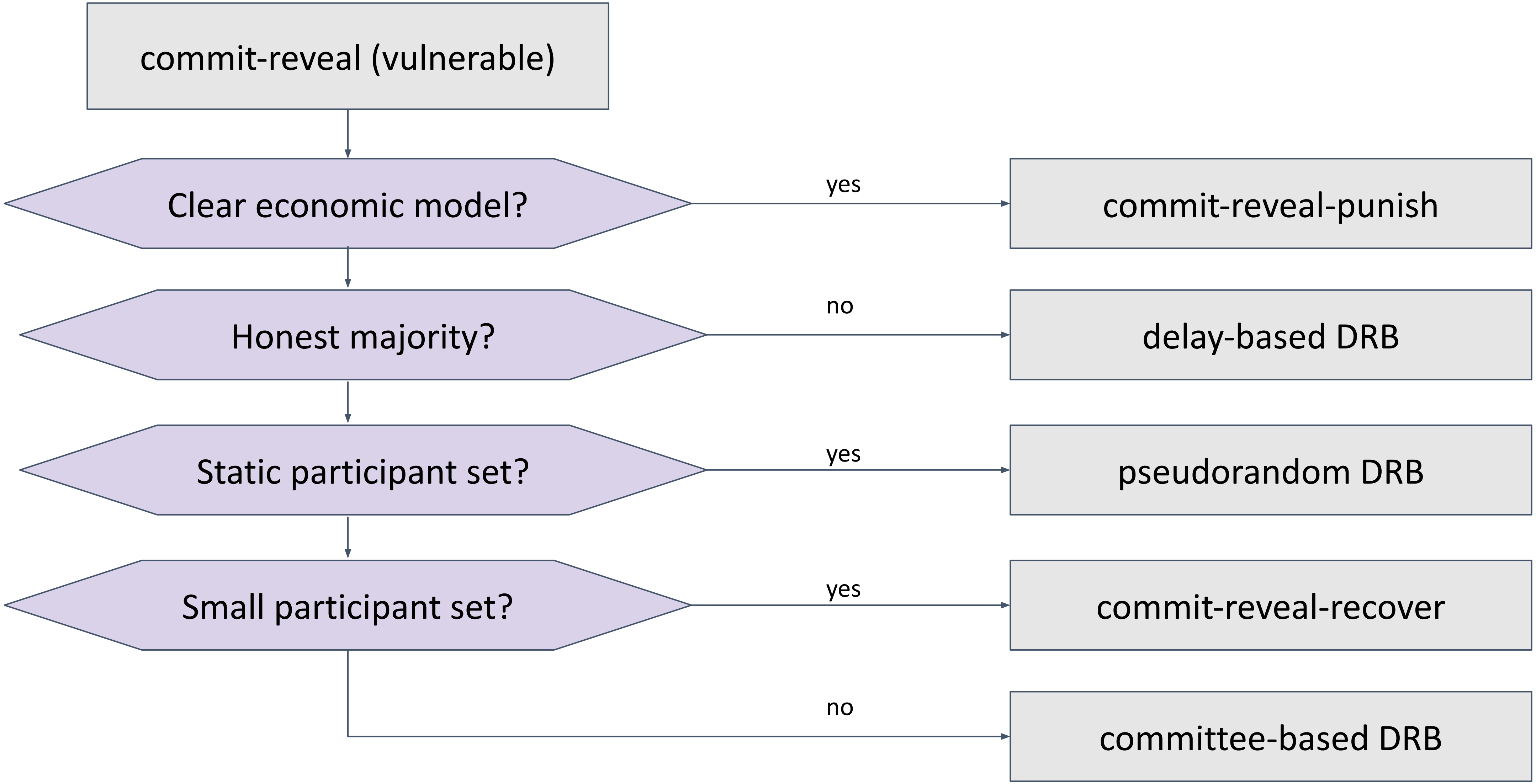
Beacon output: $\Omega = \bot$



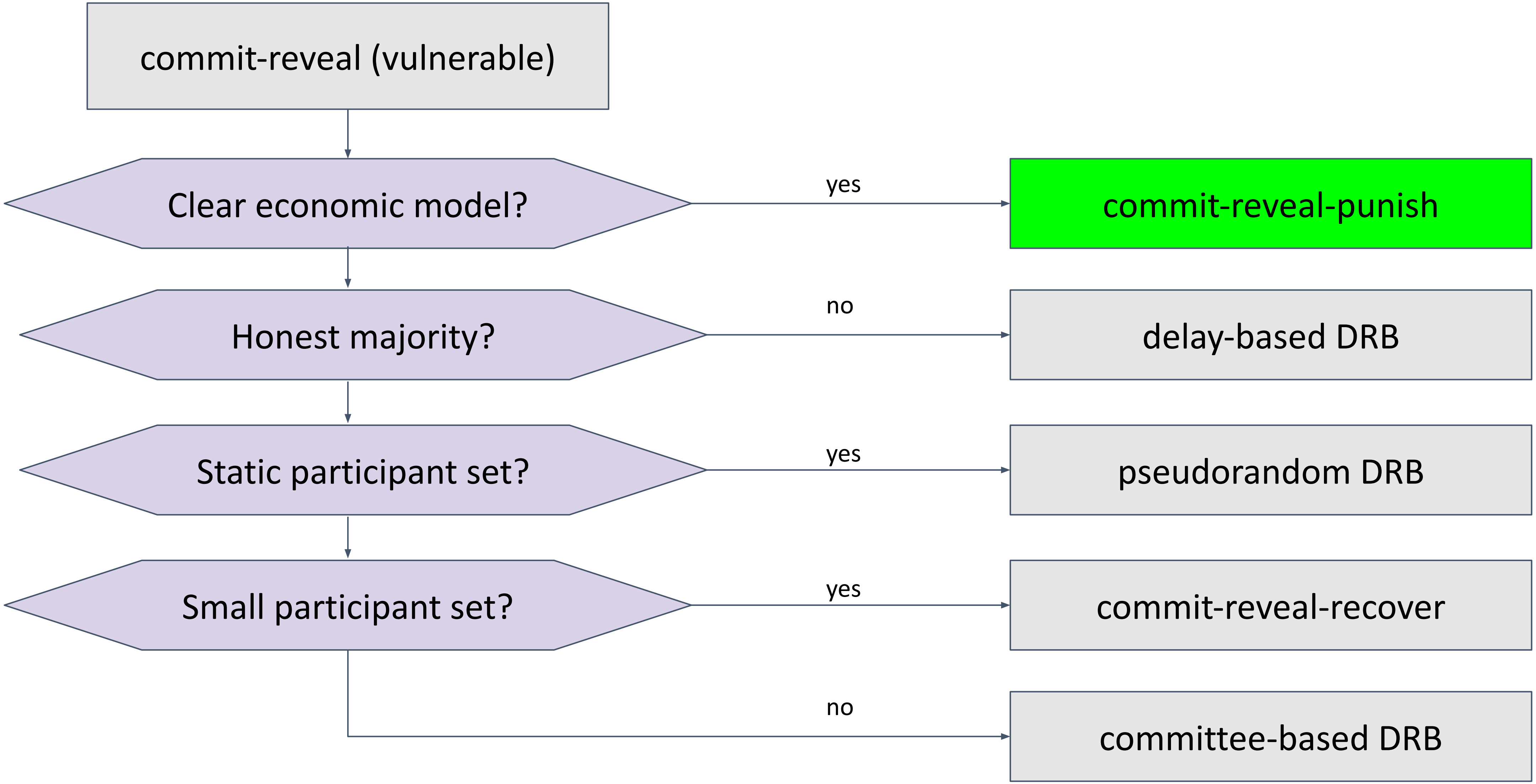
Main problem: last-revealer attack

All DRBs are essentially **patches to the last revealer attack**

DRB design flow chart

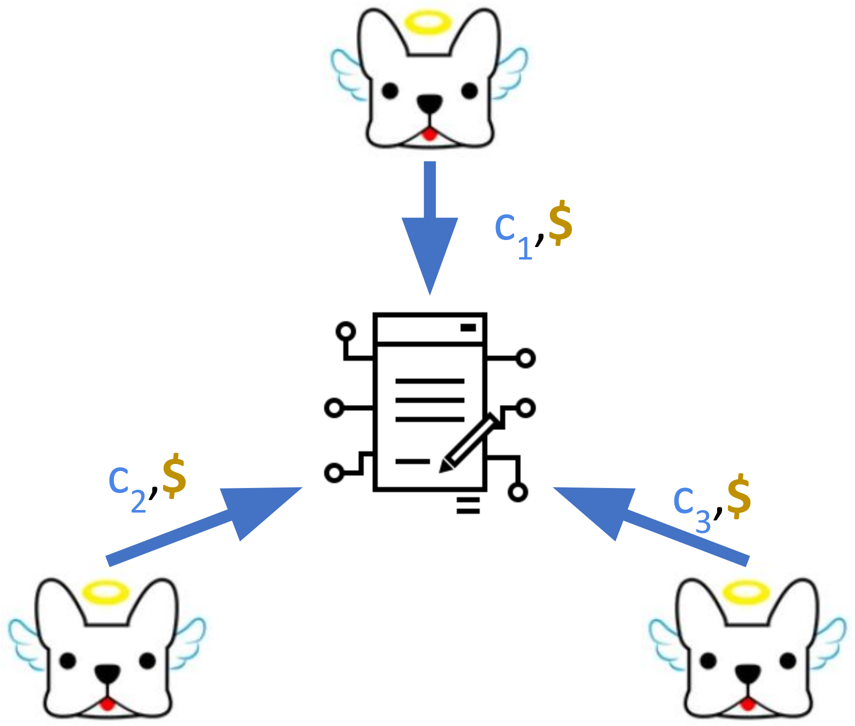


DRB design flow chart



Commit-reveal-punish

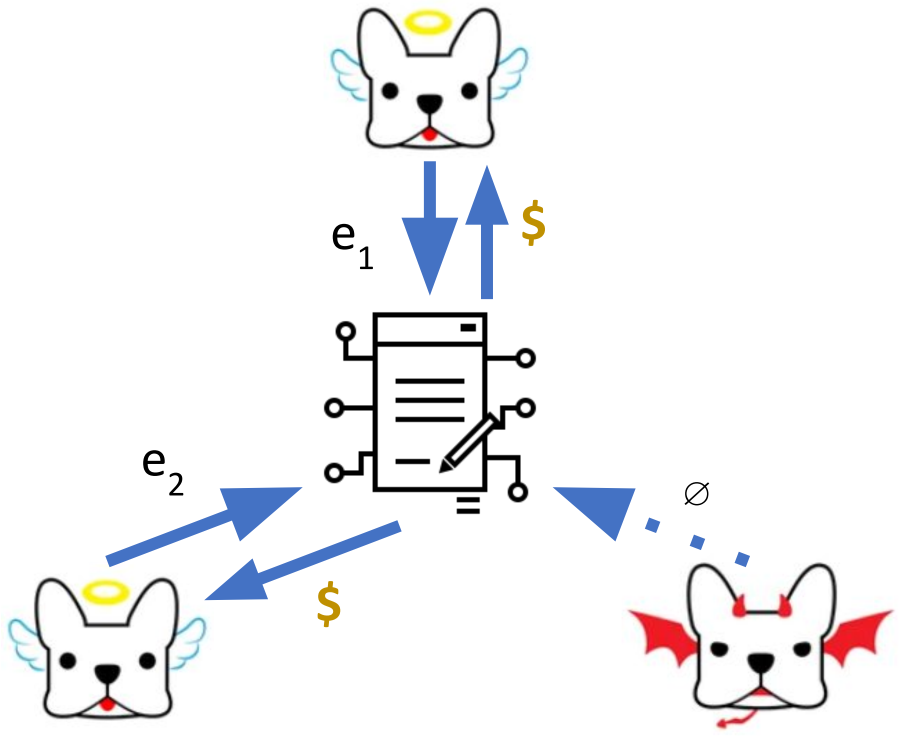
1. **Commit/deposit**



Commit-reveal-punish

1. **Commit/deposit**

2. **Reveal + refund**
   a. Participants who don't reveal lose funds
   b. Restart if any participant aborts



Commit-reveal-punish

- **Advantages**
  - efficient
    - O(n) communication, compute
  - easily implemented

- **Cons**
  - Requires capital lockup
  - Benign faults must be punished
  - Hard to bound attacker utility if beacons have multiple purposes

Commit-reveal-punish: RANDAO

ethereum

- Deployed in Ethereum since 2020
  - Used for committee selection
- Also available to smart contracts
  - `block.prevrandao`
- Proposer can reveal VRF or withhold
  - Withholding precludes block reward
- Withholding is profitable! [AW24]

Commit-reveal-punish: RANDAO

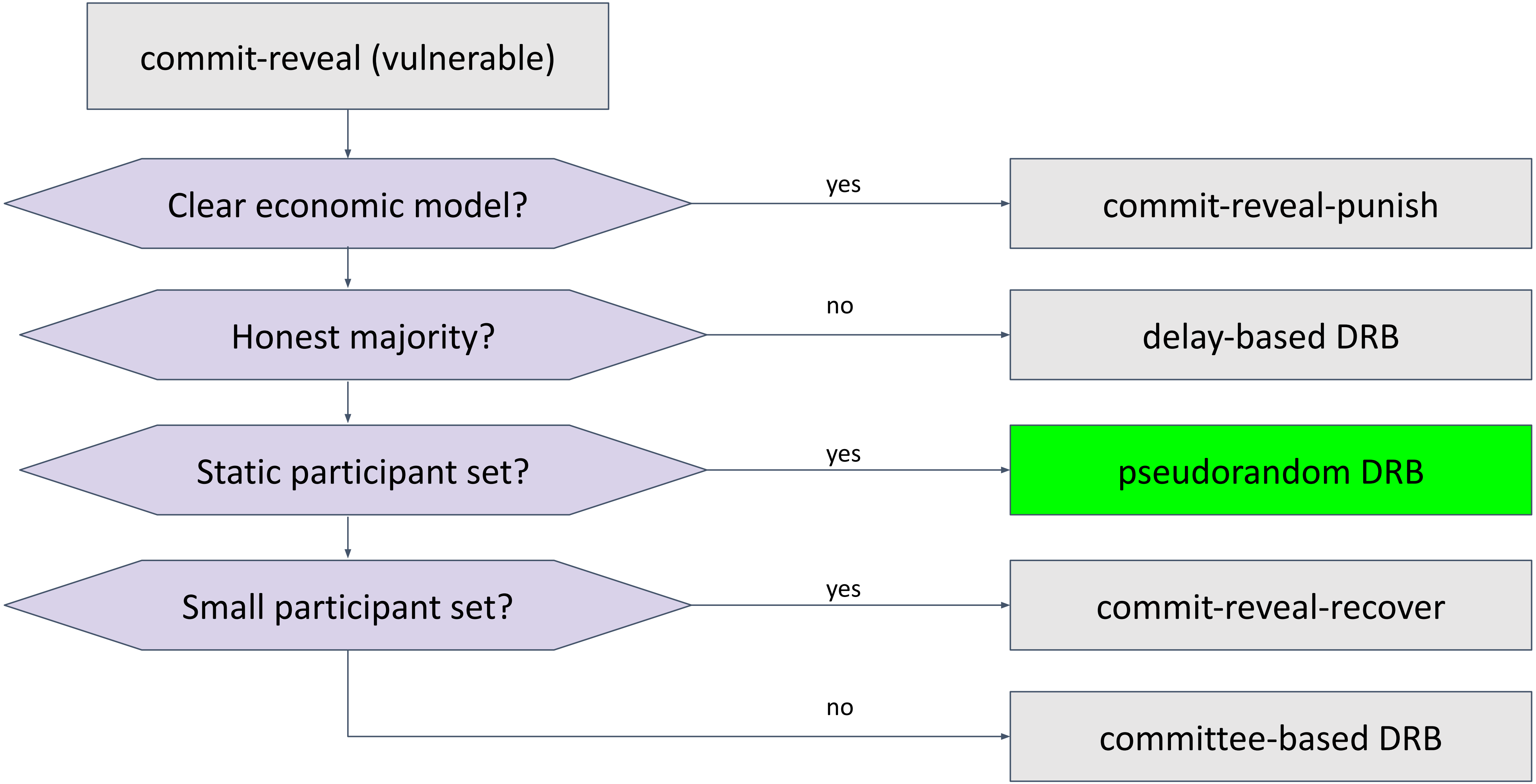
ethereum

- Deployed in Ethereum since 2020
  - Used for committee selection
- Also available to smart contracts
  - `block.prevrandao`
- Proposer can reveal VRF or withhold
  - Withholding precludes block reward
- Withholding is profitable! [AW24]

α	optimal reward
1%	1.00107%
5%	5.04834%
10%	10.18807%
15%	15.39960%
20%	20.67770%
25%	26.02472%
30%	31.45164%
35%	36.97348%
40%	42.62435%
45%	48.49184%

Optimal RANDAO Manipulation in Ethereum.
Kaya Alpturer, S. Matthew Weinberg. AFT 2024.

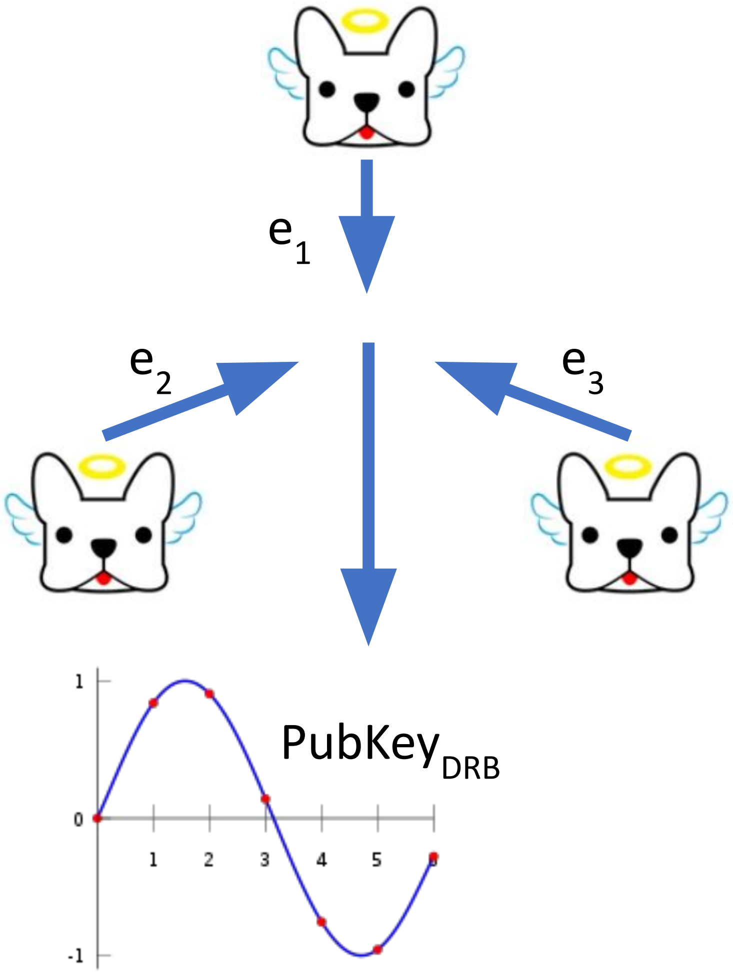
DRB design flow chart



Pseudorandom DRBs

1. **Setup**
   a. Output is *t*-out-of-*n* secret-shared VRF key
   b. Can be distributed setup (DKG) or centralized

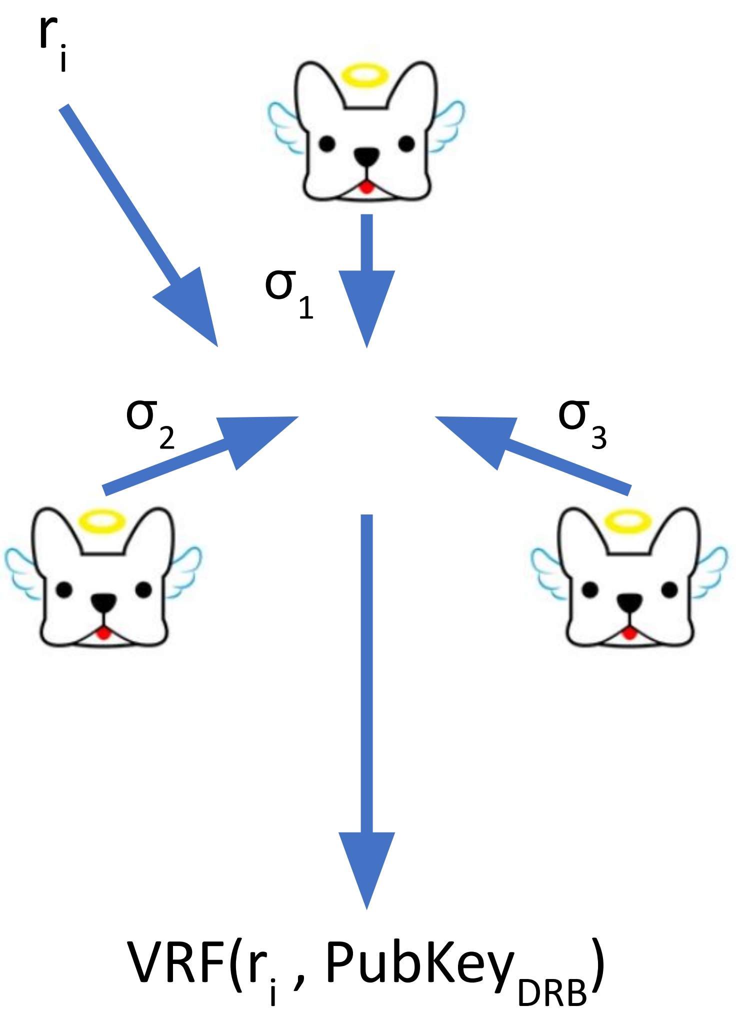
Setup itself must be a DRB!



Pseudorandom DRBs

1. **Setup**
   a. Output is *t*-out-of-*n* secret-shared VRF key
   b. Can be distributed setup (DKG) or centralized

2. **Output**
   b. Compute VRF on round input r_i
   c. Can be static (epoch number) or prior Ω
   d. Collect partial VRF evaluations
   e. Combine to output distributed VRF



Pseudorandom DRBs

- **Advantages**
  - Efficient
    - O(n) communication, compute
  - Dishonest majority cannot manipulate
    - Can only predict or stall

- **Challenges**
  - If *t* needed to compute, *n-t* can block liveness
  - Malicious coalition can predict infinitely far into the future
  - No recovery from compromise
    - Prudent to periodically re-key

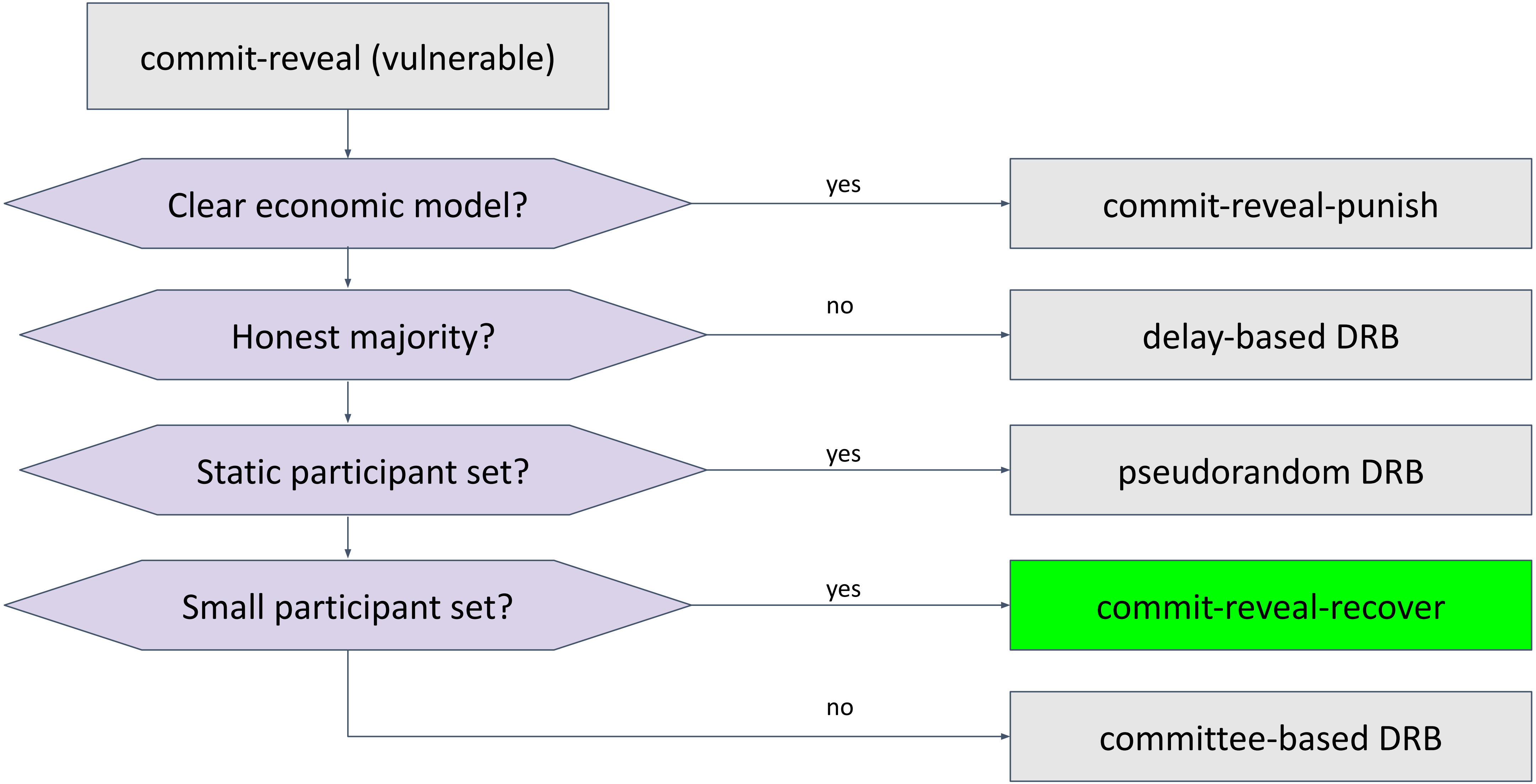
Pseudorandom DRB variants

- drand/DFINITY
  - Threshold BLS signatures
- STROBE [BCKKLNNRS 21]
  - Threshold RSA decryption, with *history generation*
- RandHerd [SJKGGKFF 17]
  - Threshold Schnorr, sharded into groups
- DDH-DRB, GLOW-DRB [GLOW 20]
  - Threshold DDH-VRF (like BLS, but NIZK instead of pairings)

drand: a production pseudorandom DRB

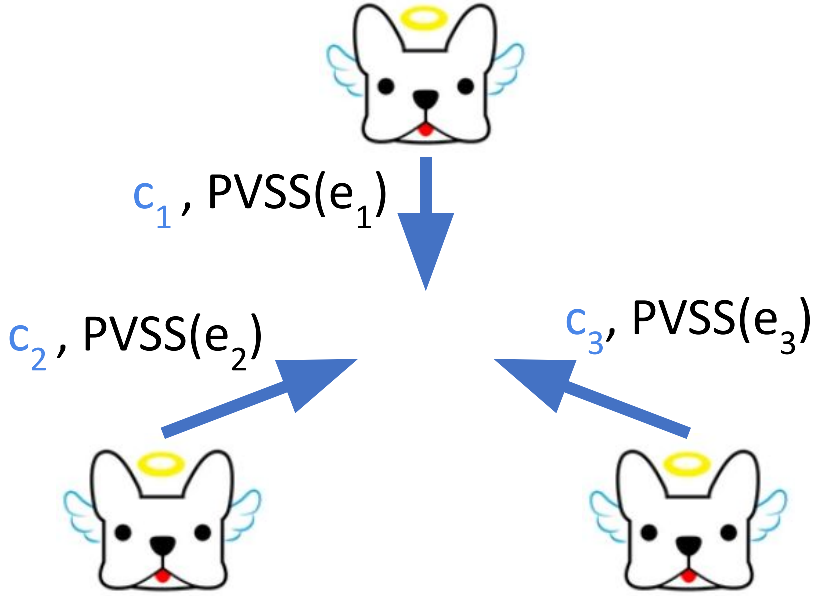
- Launched 2019
- Threshold BLS signatures
  - BLS12-381 curve
- Currently 20 nodes
  - t=11 required to sign
- 256 bits every 30 seconds
- Nodes run by academics + industry

DRB design flow chart



Commit-reveal-recover

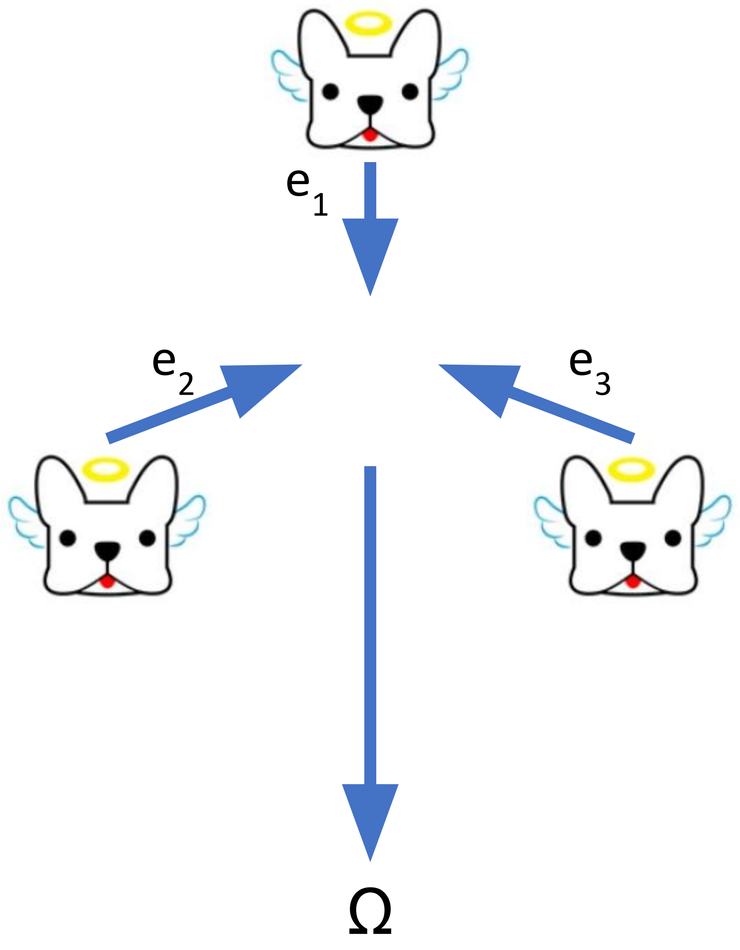
1. **Commit**
   a. Publish c_i = Commit(e_i) as in classic CR
   b. Secret-share e_i with all other nodes
      i. PVSS: Publicly verifiable secret sharing



Commit-reveal-recover

1. **Commit**
   a. Publish c_i = Commit(e_i) as in classic CR
   b. Secret-share e_i with all other nodes
      i. PVSS: Publicly verifiable secret sharing

2. **Reveal**
   b. Publish e_i as in classic CR

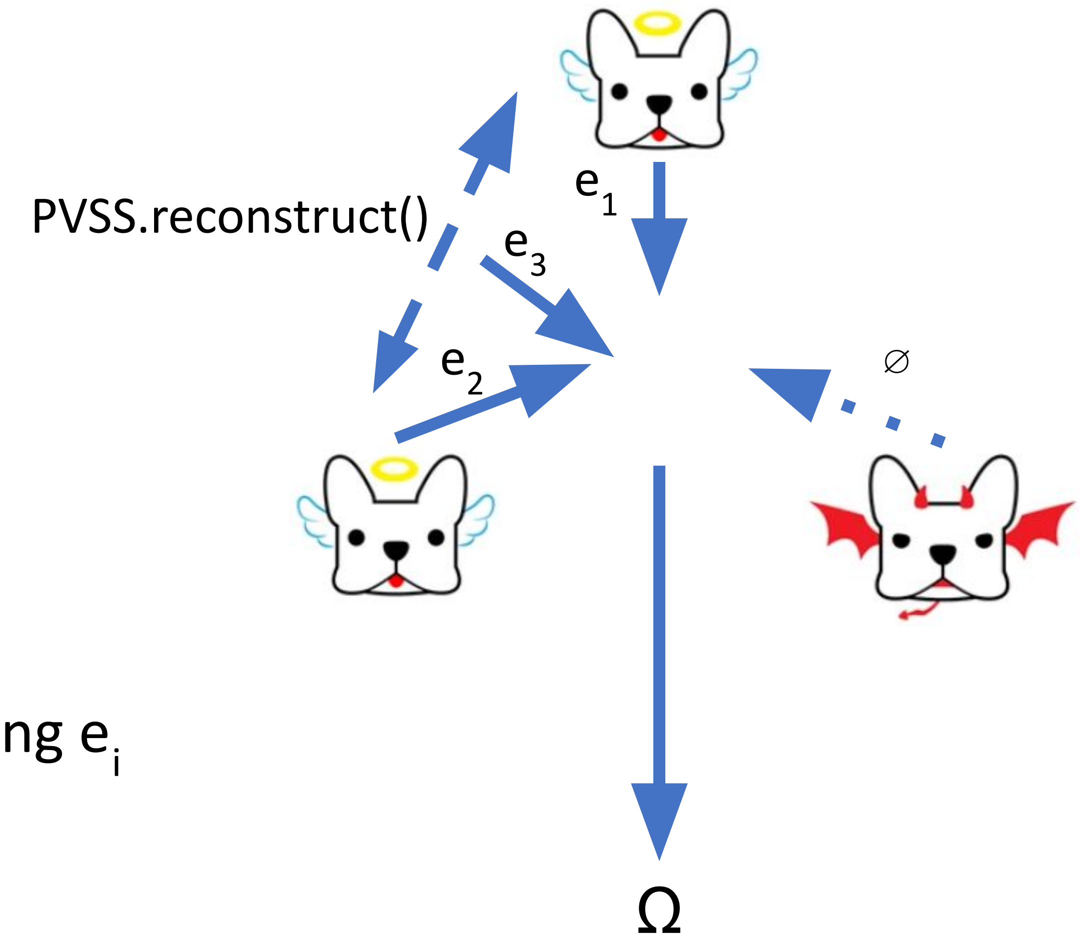


Commit-reveal-recover

1. **Commit**
   a. Publish c_i = Commit(e_i) as in classic CR
   b. Secret-share e_i with all other nodes
      i. PVSS: Publicly verifiable secret sharing

2. **Reveal**
   b. Publish e_i as in classic CR

3. **Recover**
   c. Honest participants reconstruct any missing e_i



Commit-reveal-recover variants

- Better PVSS
  - SCRAPE [CD 17]

- Amortized PVSS
  - HERB [CSO 19]
  - Albatross [CD 20]

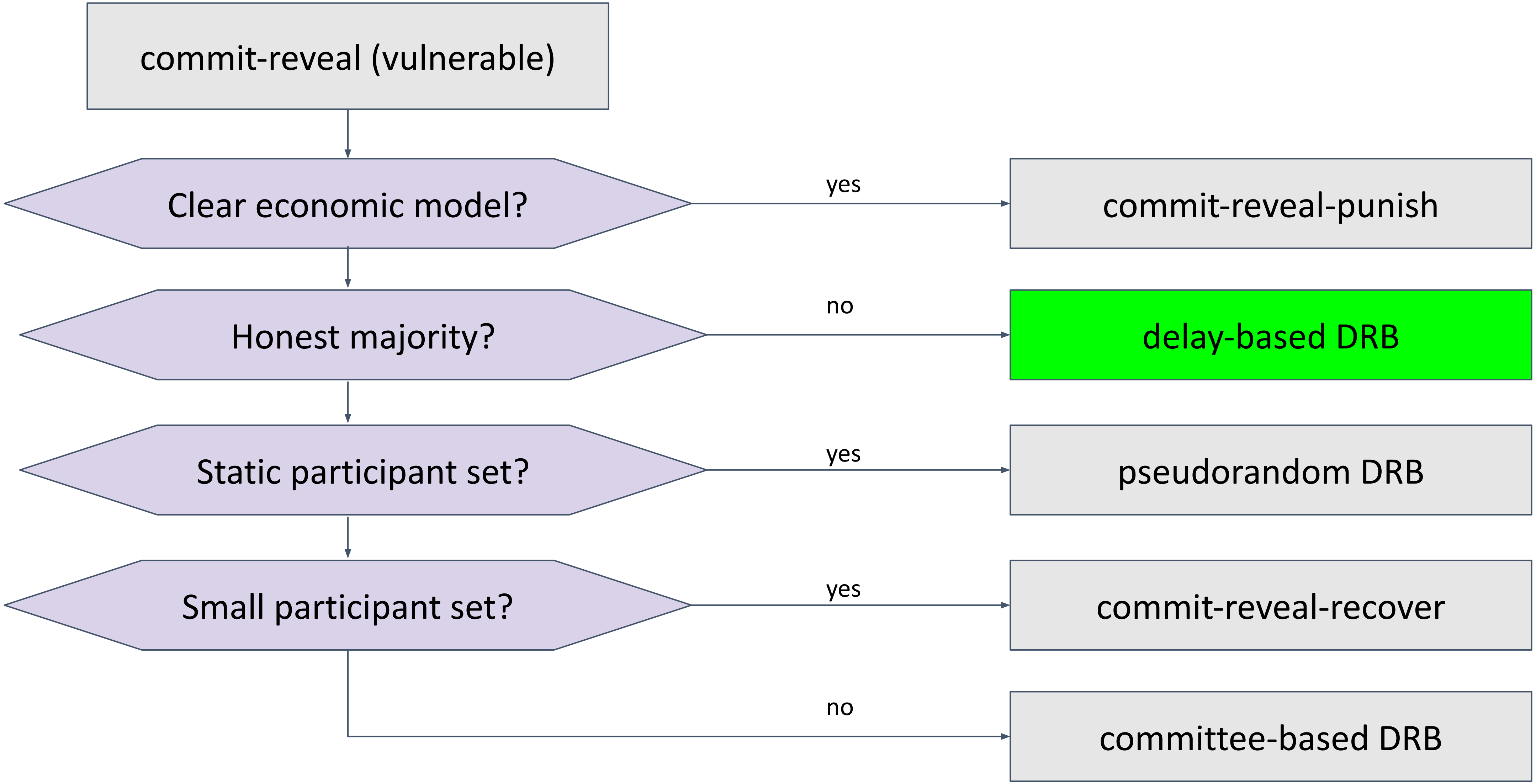
- Remove optimistic case (share-reconstruct-aggregate)
  - RandShare [SJKGGKFF 17]
  - SecRand [GSX 20]

Commit-reveal-recover

- **Advantages**
  - Flexible participation
  - Per-round entropy

- **Challenges**
  - Relatively inefficient
    - $O(n^2)$ communication
  - Complex protocols
  - If *t* needed to compute, *n-t* can block liveness
  - Reconstruction causes extra overhead

DRB design flow chart



Delay functions are *slow* (sequential) but *tractable*

Fast

Encryption
Decryption
Signing
Verification
Hashing

Delay functions:

take a *specified* number of sequential steps

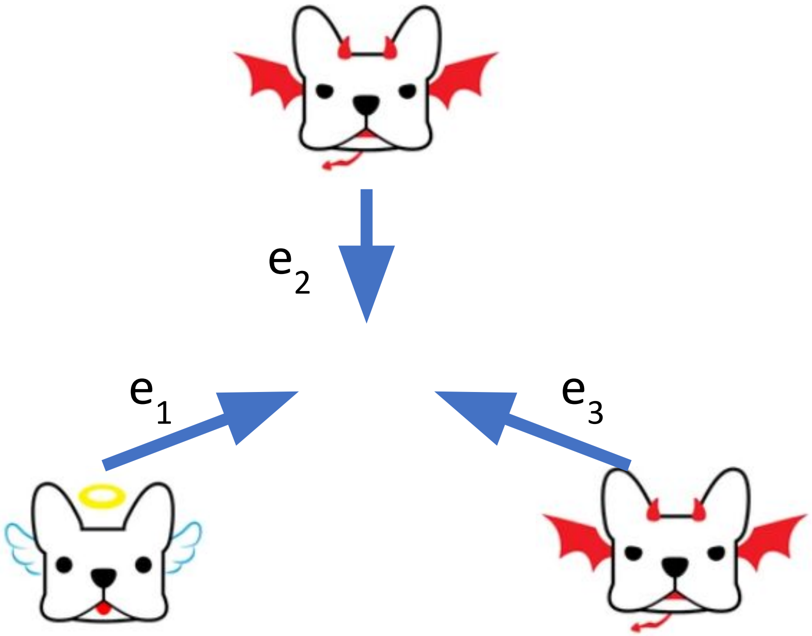
VDFs
Timed commitments
Time-lock encryption
Delay encryption
…

Intractable

Key search
Discrete log
Factoring
Collision-
findng

Reveal-delay (Unicorn) [LW15]

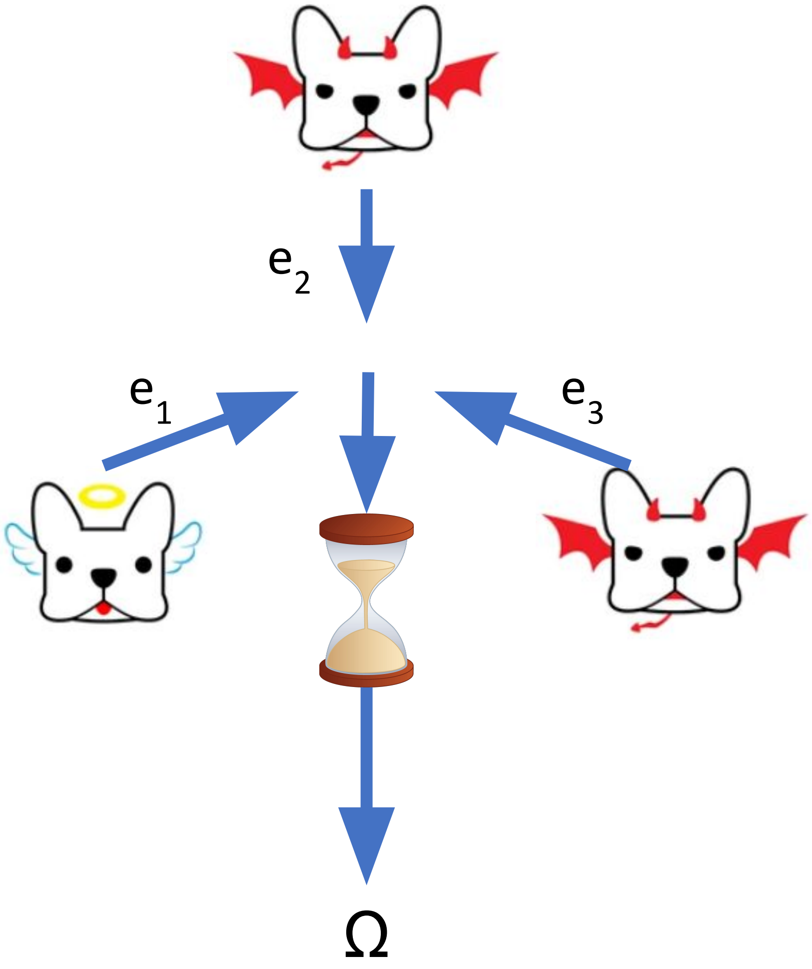
1. **Reveal**
   a. Raw entropy, no commitments needed!



Reveal-delay (Unicorn) [LW15]

1. **Reveal**
   a. Raw entropy, no commitments needed!

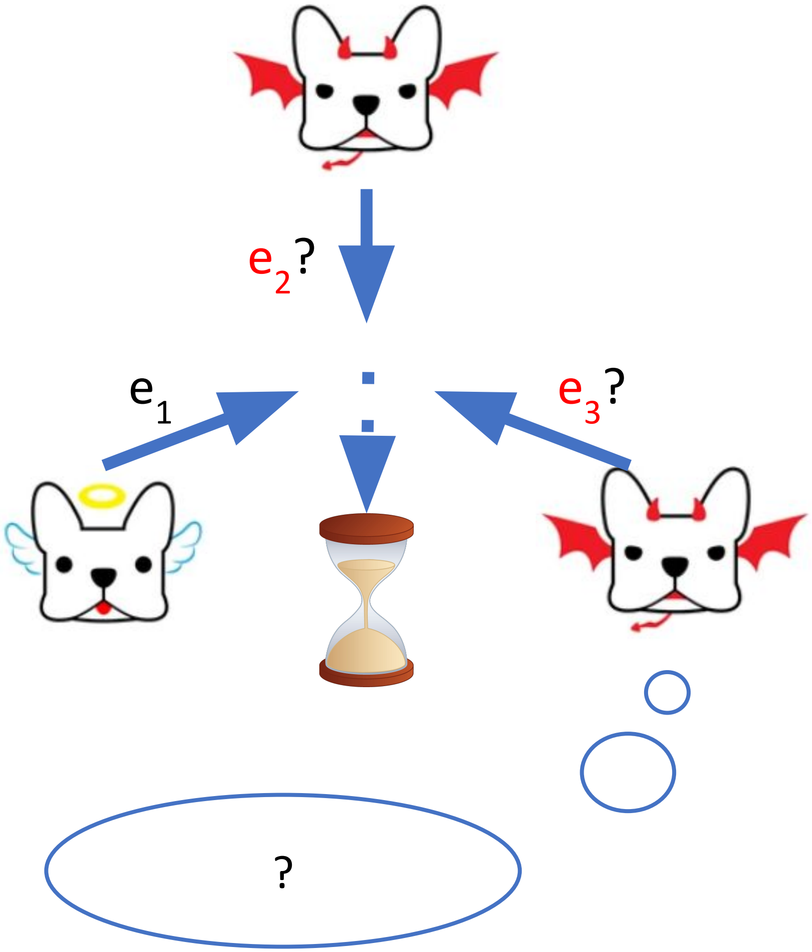
2. **Delay + combine**
   b. Modern approach: use a VDF
      i. Slow (sequential) to compute
      ii. Efficiently verifiable



Reveal-delay (Unicorn) [LW15]

1. **Reveal**
   a. Raw entropy, no commitments needed!

2. **Delay + combine**
   b. Modern approach: use a VDF
      i. Slow (sequential) to compute
      ii. Efficiently verifiable



Last revealer(s) can't compute VDF fast enough to bias

Delay-based DRB variants

- **Frequent output:** RandRunner [SJSHW20]
  - Deliver output more often than delay parameter via pipelining

- **Efficient optimistic case:** Bicorn [CATB23]
  - Skip delay function if *all* participants are honest

- **Sublinear communication:** Cornucopia [CCB24]
  - Leader gathers contributions and broadcasts succinct commitment, proofs

Delay-based DRBs

- **Advantages**
  - Secure under dishonest majority
  - Efficient
    - O(n) communication, compute. Can be reduced w/leader
  - Flexible participation
  - Per-round entropy
  - Guaranteed output delivery

Delay-based DRBs in practice: Chia

- Used for consensus
  - Launched 2021
- VDF: Repeated squaring in class group
  - 1024-bit discriminant
  - Wesolowski proofs [W19]

Delay-based DRBs

- **Advantages**
  - Secure under dishonest majority
  - Efficient
    - O(n) communication, compute. Can be reduced w/leader
  - Flexible participation
  - Per-round entropy
  - Guaranteed output delivery

- **Challenges**
  - Delay functions induce latency
  - Some party must compute the delay function
    - a public good?
  - Relatively new cryptographic assumptions
  - Intra-predictability: attacker with faster VDF may learn outcome early

VDF designs use relatively new assumptions

Cryptology ePrint Archive   Papers   Submissions   About

Paper 2024/873

Cryptanalysis of Algebraic Verifiable Delay Functions

Alex Biryukov, University of Luxembourg, Esch-sur-Alzette, Luxembourg
Ben Fisch, Yale University, New Haven, USA
Gottfried Herold, Ethereum Foundation, Bonn, Germany
Dmitry Khovratovich, Ethereum Foundation, Luxembourg, Luxembourg
Gaëtan Leurent, INRIA, Paris, France
María Naya-Plasencia, INRIA, Paris, France
Benjamin Wesolowski, CNRS, ENS Lyon, Lyon, France

Abstract

Verifiable Delay Functions (VDF) are a class of cryptographic primitives aiming to guarantee a minimum computation time, even for an adversary with massive parallel computational power. They are useful in blockchain protocols, and several practical candidates have been proposed based on exponentiation in a large finite field: Sloth++, Veedo, MinRoot. The underlying assumption of these constructions is that computing an exponentiation x^e requires at least $\log_2 e$ sequential multiplications.

Metadata

Available format(s)

PDF

Category

Attacks and cryptanalysis

Publication info

A minor revision of an IACR publication in CRYPTO 2024

Keywords

Verifiable Delay Functions, MinRoot, Veedo, Sloth++, cryptanalysis, smoothness

Contact author(s)

Theorem: dishonest majority DRBs require delay!

- **Classic result:** Dishonest majority DRBs impossible in "plain model"
  - *Limits on the security of coin flips when half the processors are faulty*. Richard Cleve. TOC 1986.

- **Practical observation**: Dishonest majority DRBs possible with VDFs

- **New result**: Dishonest majority DRB *require* delay functions!
  - *Good Things Come to Those Who Wait: Dishonest-Majority Coin-Flipping Requires Delay Functions*. Joseph Bonneau, Benedikt Bünz, Miranda Christ, Yuval Efron. Eurocrypt 2025.
  - Simple delay function, not full VDF
    - Not parameterizable, not efficiently verifiable
  - Assumes network synchrony

Open questions (protocol design)

- *Secret Leader Election*
  - DRBs where only winner founds out they have won!
  - Applications in consensus and other protocols
- *Silent setup*
  - Use existing public keys for threshold DRB with no (or limited) setup phase
- *Optimistic protocols*
  - Faster execution if a chosen leader is honest
  - Faster execution if *all* nodes are honest

Open questions (engineering)

- Simpler API for developers
  - Smart contract integration: Aptos Roll, Mysten sui::random
- VDF deployment
  - Security model requires public access to VDF hardware
- Local randomness generation
  - DRB is no better than nodes' RNGs!
- Public trust

Thank you!

For more, please see 3 derailed surveys:

SoK: Decentralized randomness beacon protocols. Raikwar, Mayank, and Danilo Gligoroski. ACISP 2022. https://arxiv.org/pdf/2205.13333

SoK: Distributed Randomness Beacons. Kevin Choi, Athira Manoj and Joseph Bonneau. IEEE S&P 2023. https://eprint.iacr.org/2023/728

SoK: Public Randomness. Kavousi, Alireza, Zhipeng Wang, and Philipp Jovanovic. EuroS&P 2024. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10629002

Backup slides

Is a dishonest majority model worthwhile?

- Consensus requires an honest majority...
  - **Counterpoint:** Attacks on DRBs are *invisible*

- Dishonest majority enables *open participation*
  - No security downside to adding more participants!