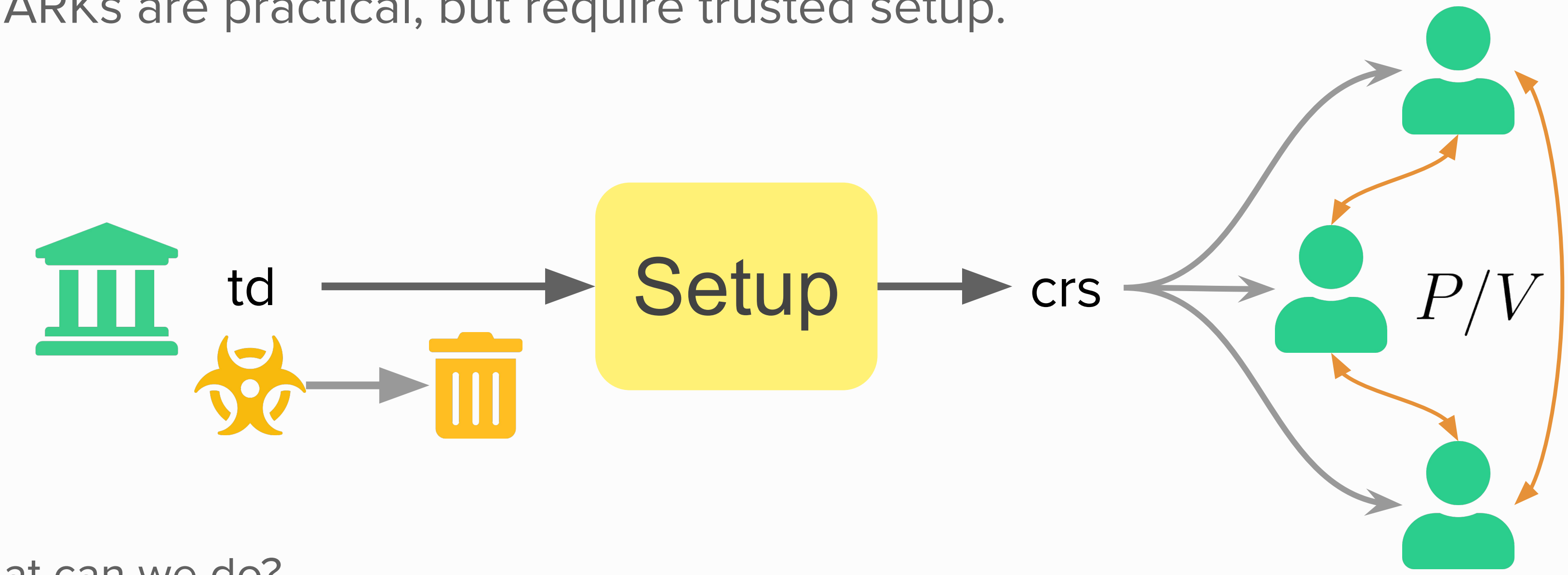
Framework for Snarky Ceremonies

Markulf Kohlweiss[1,2], Mary Maller[3], Janno Siim[4], Mikhail Volkhov[2]

1. IOHK
2. The University of Edinburgh, UK
3. Ethereum Foundation
4. The University of Tartu, Estonia

CRS and Public Setup

SNARKs are practical, but require trusted setup.



What can we do?

- Designated verifier generates an SRS
- MPC
- Subversion resistance (soundness, ZK)
- Updatable or universal SRS
- Use RO => transparent solutions

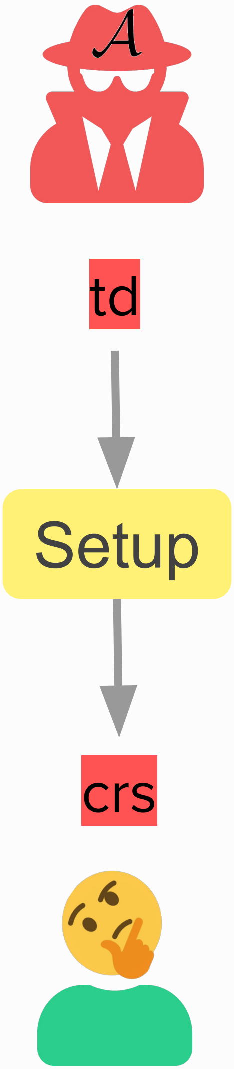
Subversion Security

Any security left even if CRS is compromised? Yes! Somewhat!

- S-X: the scheme achieves X even if CRS is bad.
- One can have S-Soundness or S-ZK [BFS16]
- *But not S-Soundness + ZK*

In practice:

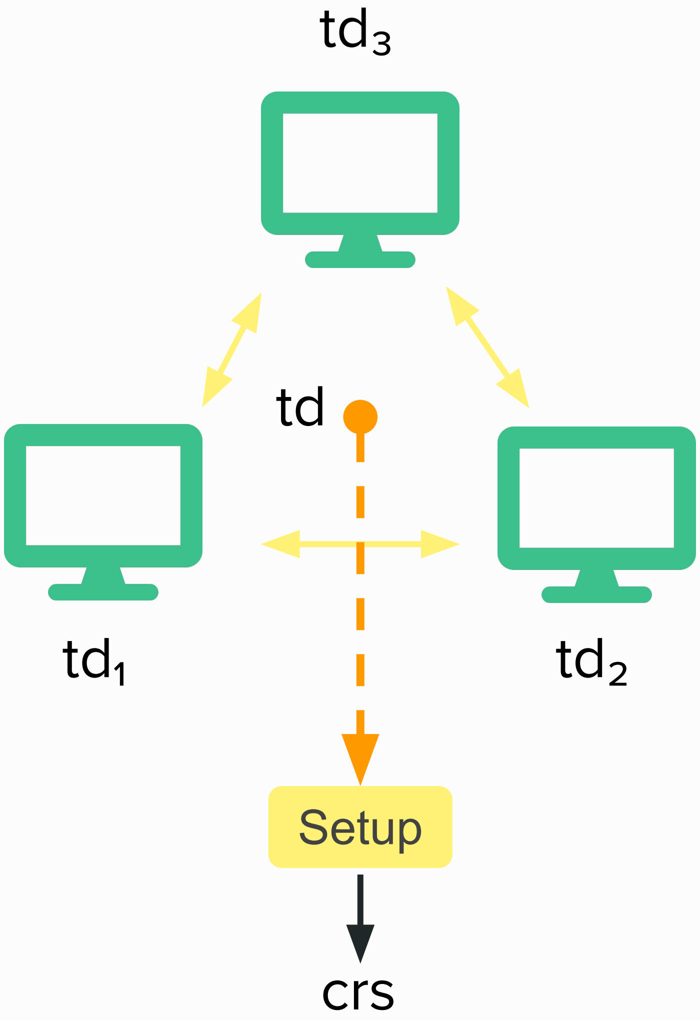
- Subversion-ZK is not expensive
  - Groth16 can achieve S-ZK. [ABLZ17]
  - Also Groth-Maller17, Sonic, …



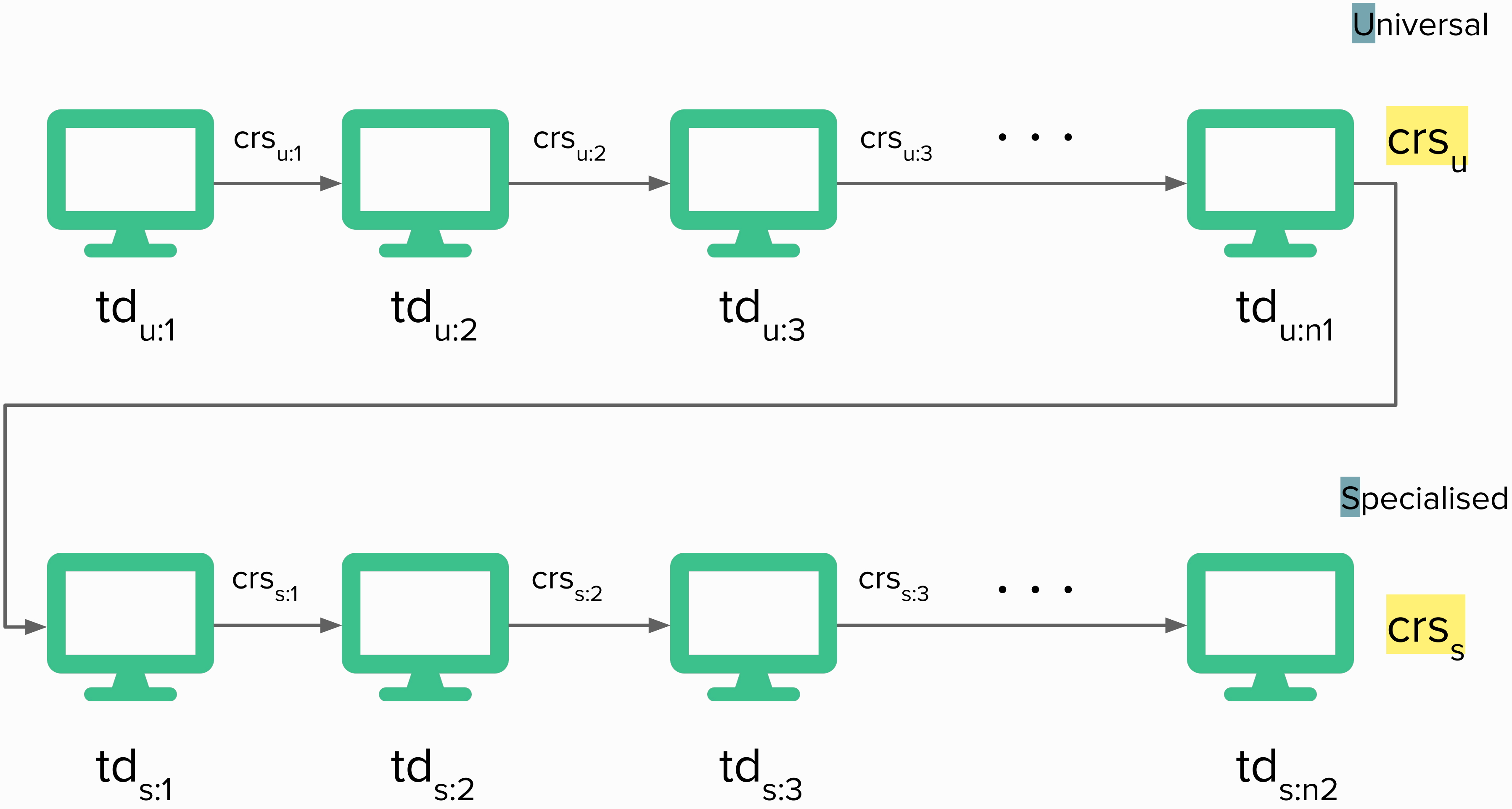
MPC and variants

1. [BCGTV15] “Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs”
    - Generic
    - Pre-commitment phase, requires parties’ avaliability
    - [BGG15] Bowe-Gabizon-Green
        - Instantiate (1) with Pinnochio
        - Sub ZK
    - [ABLSZ19] “UC-Secure CRS Generation”
        - UC Modelling of (1), for Groth16
2. [BGM17] Bowe-Gabizon-Miers
    - For Groth16
    - Player-exchangeable
    - But random beacon
    - 2 phases; first, universal, called “Powers of Tau”

All protocols: at least one party should be honest



BGM17 Protocol



Final CRS: (crs_u, crs_s)

Random Beacon

What:

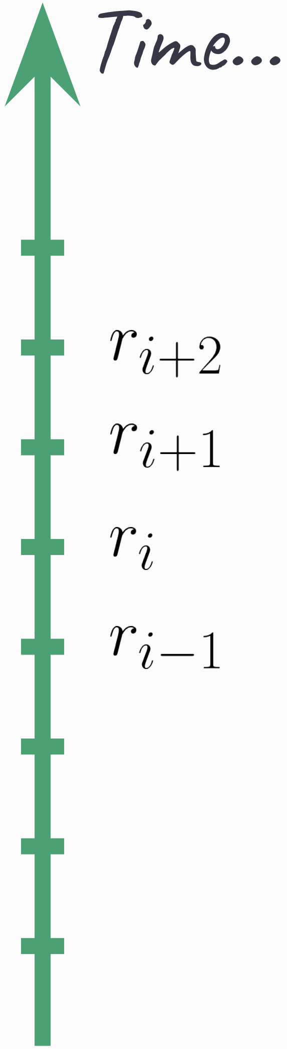
- Periodical unpredictable randomness. Public, verifiable, unbiased.

How to construct:

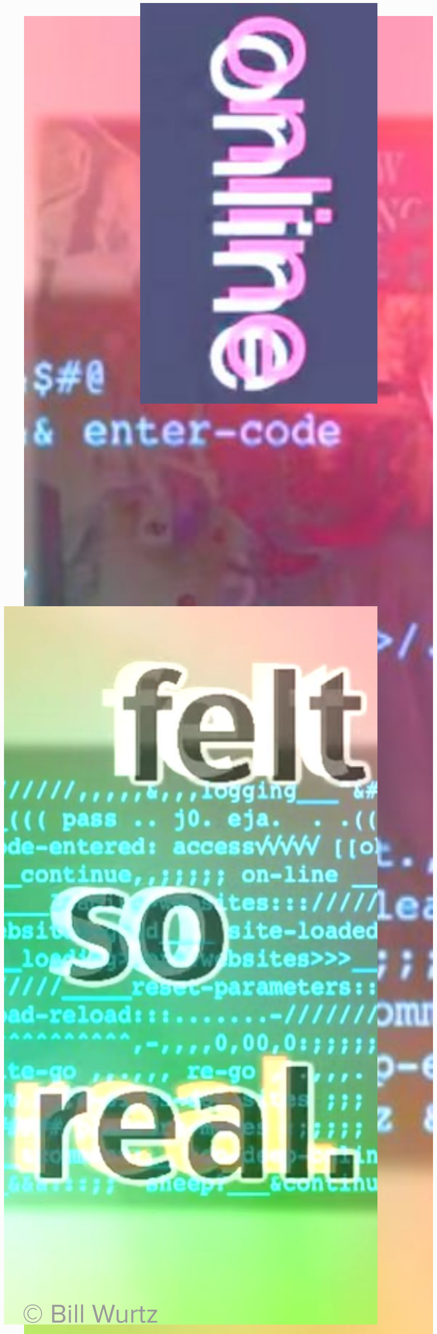
- Apply a VDF (verifiable delay function) to a public source of entropy.
- E.g. hash bitcoin block many times

How to apply:

- RB is a last “participant” of each phase.
- “Unbiases” the CRS



$$RB(i) = r_i$$



© Bill Wurtz

Real-world experience

[BCTV14]: ZCash Sprout MPC, 2016

- For [BCTV14], a modification of Pinocchio
- 6 participants

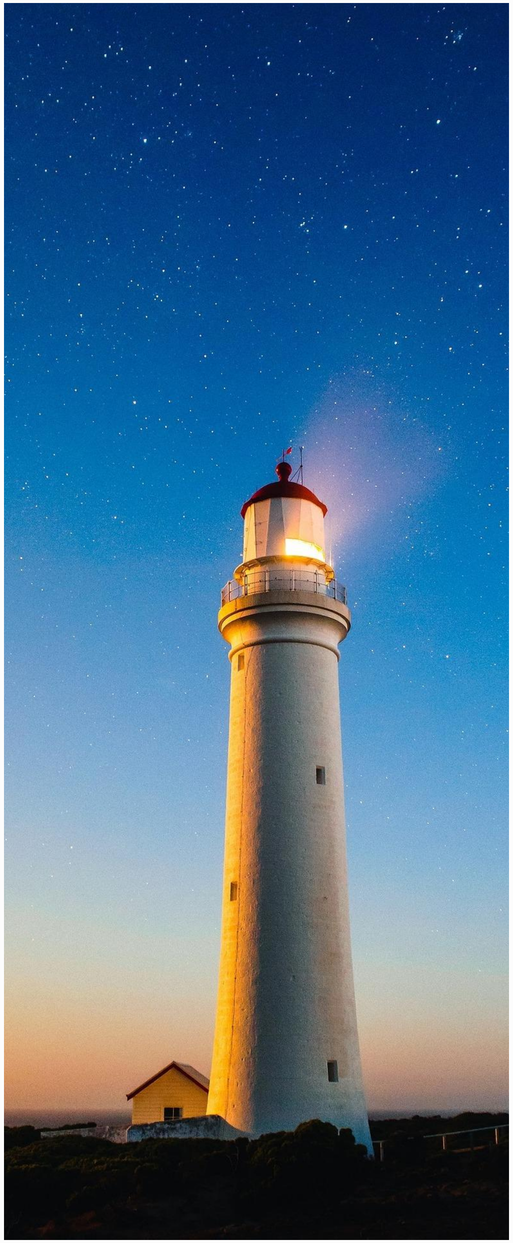
[BGM17]:

- ZCash Sapling MPC, 2017-18
  - ~90 participants in each phase, BLS12-381
- Perpetual Powers of Tau (PPoT), since 2019
  - First phase only: BN254: 70 participants, BLS12-391: 18 participants
- Filecoin, Semaphore, Loopring, Tornado Cash, Hermez
  - All based on PPoT
- Celo/Plumo

Also: Aztec Ignition (BN254, 176 participants)

Random Beacons in practice

- Bitcoin + SHA256
  - The ceremony of ZCash (2^42 SHA256), Loopring
  - Expensive to verify
- ETH + class based hidden group order VDF
  - Semaphore
  - What are the security assumptions?
- RB protocols
  - DRAND used by Hermez, HERB, Dfinity's RB, SPURT, ...
- Ignore it:
  - Filecoin
  - The draft by Mary Maller, 2018, shows that in GGM RB is not necessary; this is a starting point of our work.

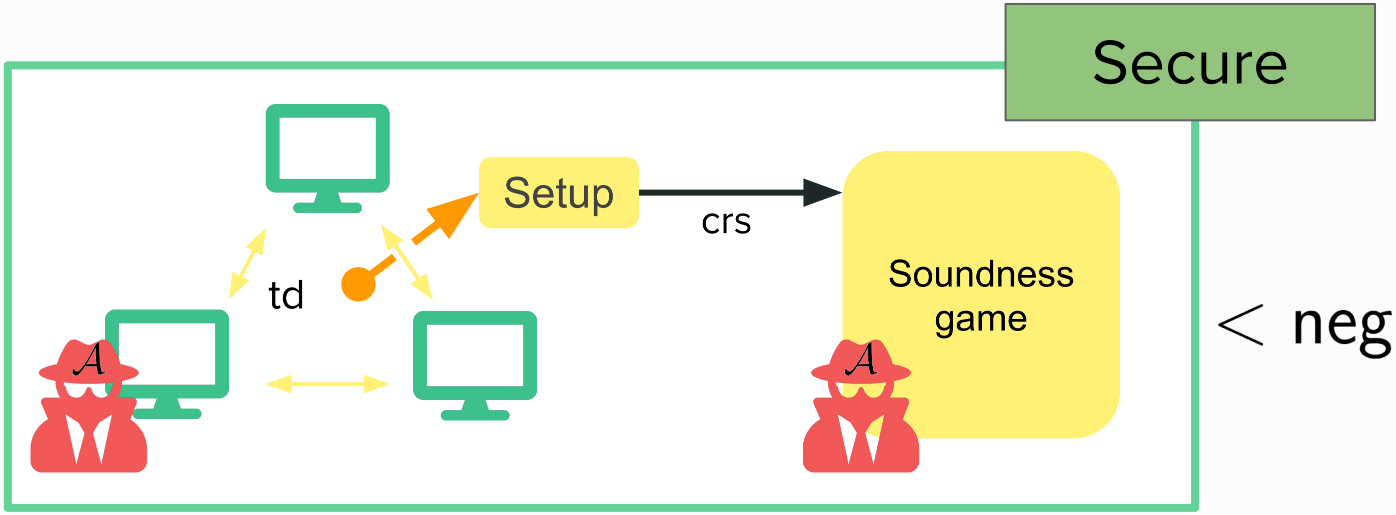


Ceremonial SNARKs / SNARKy Ceremonies

Based on our recent work [KMSV21]:

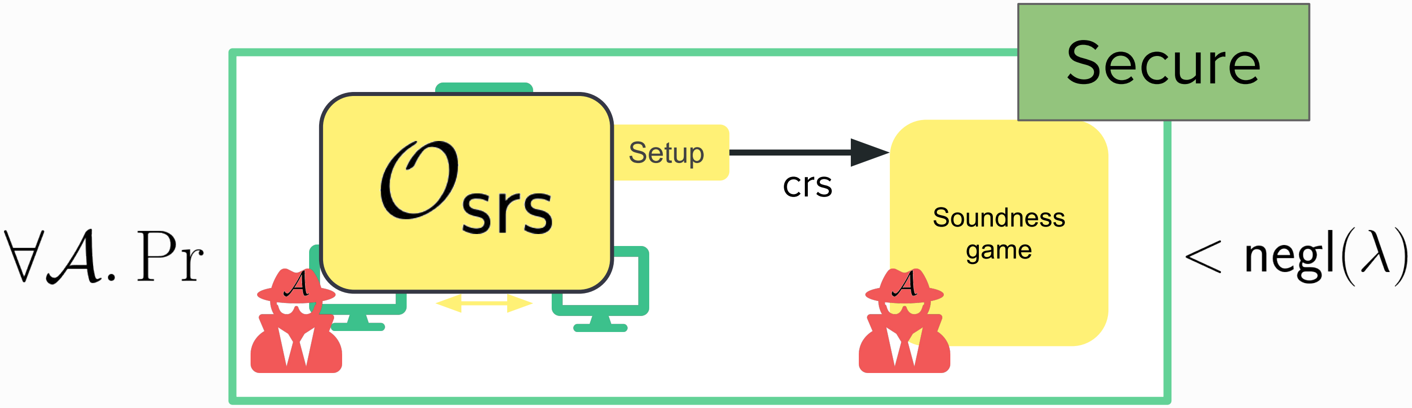
- Holistic security framework that models soundness within MPC
    - Less restrictive; does not require simulatability
- Groth16+[BGM17] proof in this framework
    - AGM+RO under q-dlog
- *Without* relying on random beacons!
    - + simplify the protocol slightly
    - + independent verification tool being developed by GRNET

$\forall \mathcal{A}.\, \Pr$



$< \mathrm{negl}(\lambda)$

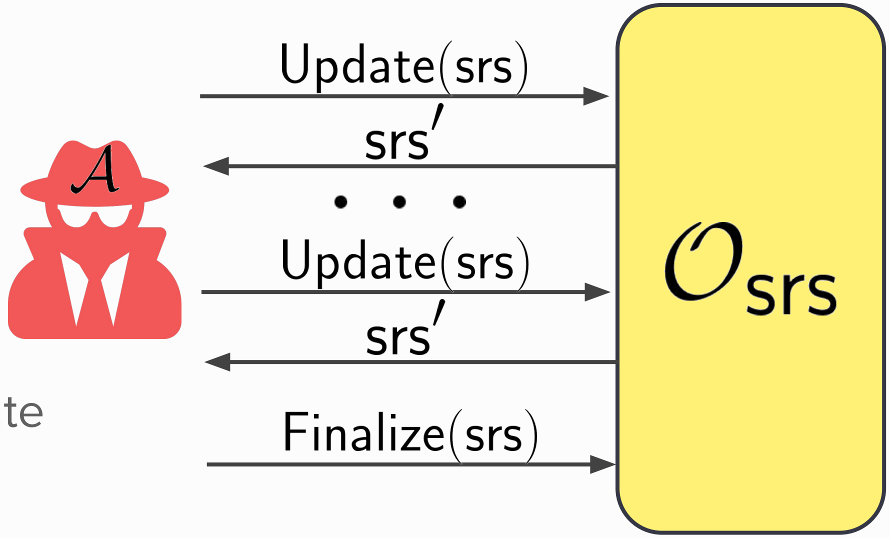
Update Knowledge Soundness



$\forall \mathcal{A}.\ \Pr$
$\mathcal{O}_{\mathsf{srs}}$
Setup
crs
Soundness game
$\mathcal{A}$
$\mathcal{A}$
Secure
$< \mathsf{negl}(\lambda)$

The SRS oracle models setup MPC:

- Multiple phases; we use 2
- In each phase: Update/Finalize
- Oracle rejects all invalid SRSs
- In each phase at least 1 honest update



$\mathcal{A}$
$\mathsf{Update}(\mathsf{srs})$
srs'
. . .
$\mathsf{Update}(\mathsf{srs})$
srs'
$\mathsf{Finalize}(\mathsf{srs})$
$\mathcal{O}_{\mathsf{srs}}$

Theory vs Practice

- Theoretically we understand ceremonies fairly well
- Significant amount of practical knowledge is accumulated
- What can we improve?
    - Although ceremonies are very transparent, they are heterogeneous and non-trivial to verify independently.
    - Can we make verification even simpler?
    - Best practices help with automatization, etc.

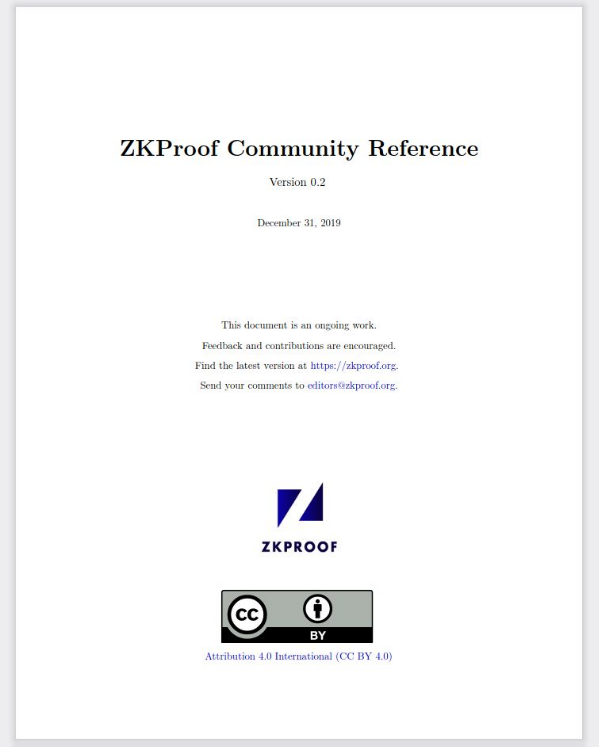
Implementations and ease of verification

- Mostly independent implementations of particular ceremonies:
  - ZCash, Tornado Cash, Aztec, Filecoin, …
- Kobi Gurkan's `phase2-bn254` repo
  - Perpetual Powers of Tau
  - Used/forked by Semaphore, Loopring, Celo
- GRNET’s independent verifier, work-in-progress

Standardization: Current status

What is already included into the Community Reference v0.2?

- `1.6.7`: Examples of setup and trust
    - Trustless/CRS separation
    - Mentions some ways to reduce trust in the CRS case, including MPC
- `3.6.2`: SRS generation
    - “Real world social and technical problems”
    - Mentions S-ZK, MPC generation, RBs, first phase reusability
    - For MPC, highlights practical and security concerns



ZKProof Community Reference

Version 0.2

December 31, 2019

This document is an ongoing work.

Feedback and contributions are encouraged.

Find the latest version at https://zkproof.org.

Send your comments to editors@zkproof.org.

ZKPROOF

Attribution 4.0 International (CC BY 4.0)

Topics and Problems

- Random beacons
- Transparent setups, pros/cons
- Simpler verification
- Ease of comparison of ceremonies
- What can we learn from past ceremonies?

Discussion points 1/2

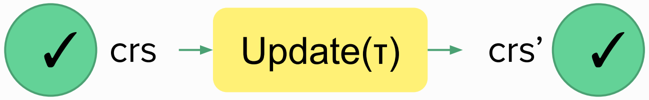
1. To what degree do we want to support/deprecate legacy ZK systems?
   a. e.g. Pinocchio, Groth16, non-transparent, non post-quantum?
   b. Should we actively discourage certain practices?
2. Standardization of cryptographic protocols/definitions for ceremony SNARKs.
   a. Our multi-phase updatable definition, subversion zero-knowledge, but also UC-type definitions (e.g, mining for privacy, [ABLZ17]).
3. Consistent documentation for execution of ceremonies.
   a. Which parameters are important (# of participants, curve, random beacon, ...)?
   b. Do we want to agree on a form/checklist for projects to fill in for README.md?
   c. Should we provide a standardised reference implementation for ceremony verification?

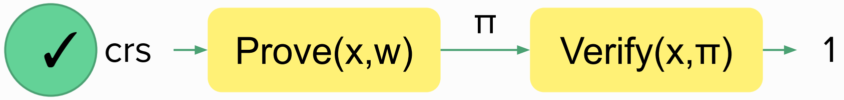
Discussion points 2/2

4. Reviewing past and ongoing ceremonies and their security models?
5. Do we need a standard for public entropy contributions and random beacons?
    - 5.1. To what extent should we rely on random beacons?
6. Do we want a common framework for one-phase and two-phase updatability?
7. Why are current practices so diverse? What are advantages of being informed by formal security analysis?

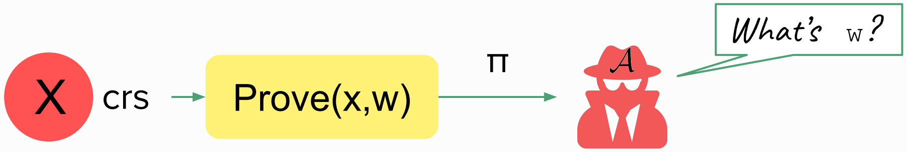
Our definitions

1. Perfect Completeness:
   - Update Completeness: correct updates pass CRS verification
   - Prover Completeness: proofs for a correct CRS pass NIZK verification





2. Subversion Zero-Knowledge
   - ZK holds for every adversarially-generated CRS that verifies



3. Update Knowledge Soundness:
   - KS holds for every CRS generated in an MPC with at least one honest participant in each phase.

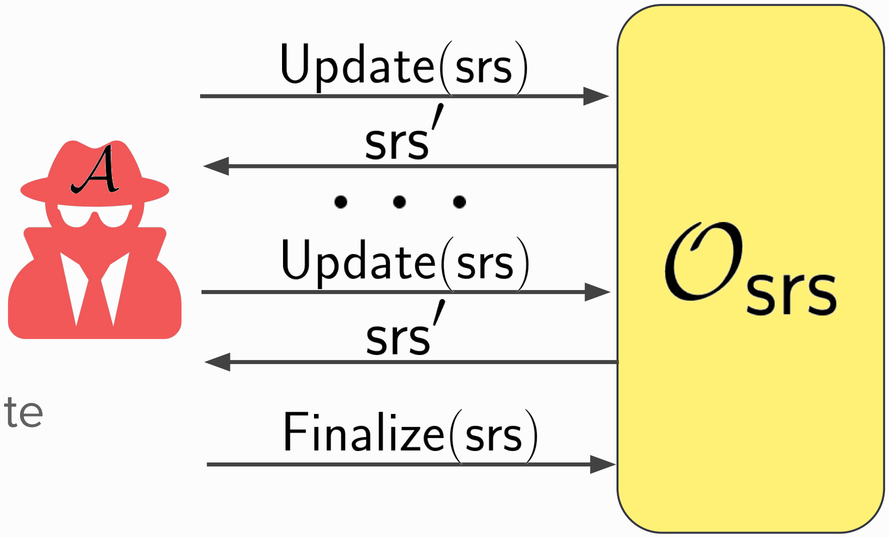
Update Knowledge Soundness

Adversary sets SRS using the oracle before attempting the forgery.

$$\left[\begin{array}{l}(\phi,\pi)\leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{srs}}(\cdot)}(1^\lambda);\ \mathit{get}\ (\mathsf{srs},\varphi)\ \mathit{from}\ \mathcal{O}_{\mathsf{srs}};\ w \leftarrow \mathcal{E}_\mathcal{A}(\mathsf{view}_\mathcal{A});\\ \boldsymbol{\mathit{return}}\ \mathsf{Verify}(\mathsf{srs},\phi,\pi)=1\wedge(\phi,w)\notin\mathcal{R}\wedge\varphi>\varphi_{max}\end{array}\right]$$

The SRS oracle models setup MPC:

- Multiple phases; we use 2
- In each phase: Update/Finalize
- Oracle rejects all invalid SRSs
- In each phase at least 1 honest update



$\mathcal{A}$
Update(srs)
srs'
. . .
Update(srs)
srs'
Finalize(srs)
$\mathcal{O}_{\mathsf{srs}}$