KU LEUVEN

CBT Workshop, Affiliated with EUROCRYPT 2026
Rome, Italy, 9th May 2026

Trust, But Verify

When Using the Powers of Tau

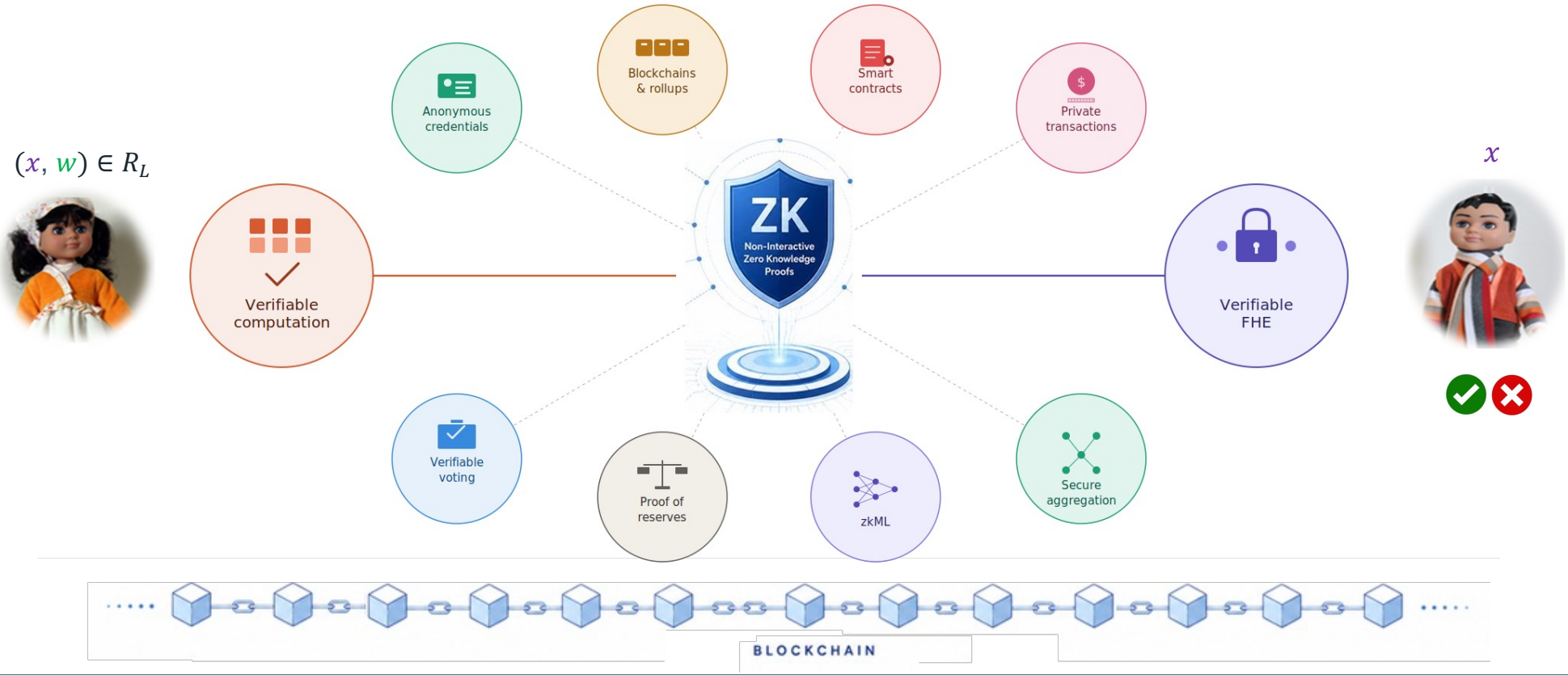
Karim Baghery

COSIC · KU Leuven

ia.cr/2025/2000

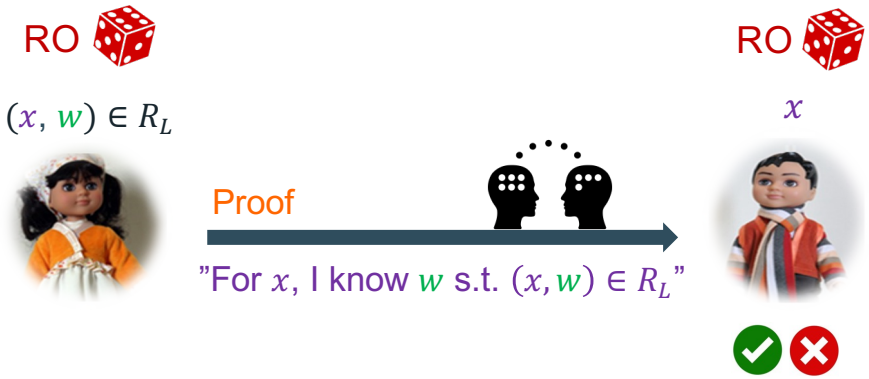
COSIC

Motivation: Applications of (Non-Interactive) ZK Proofs



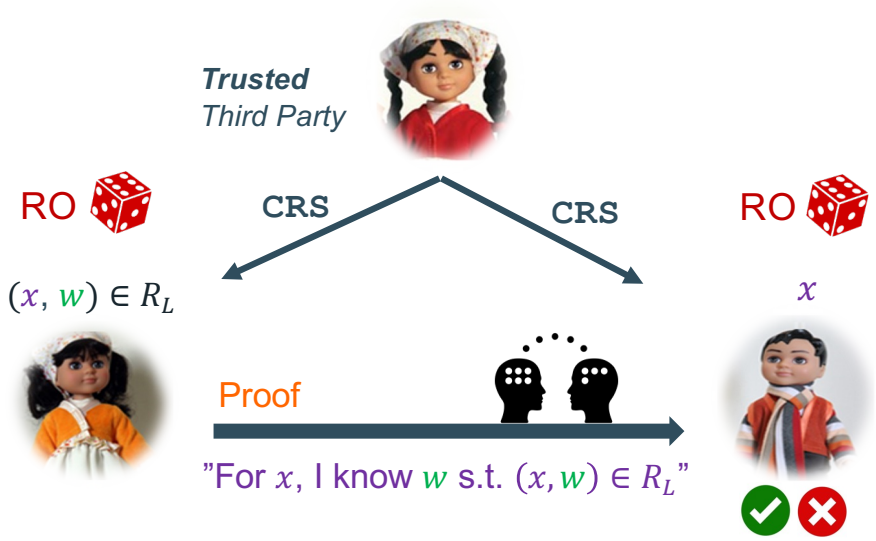
Constructing Non-Interactive Proof Systems: RO & CRS

- **Non-Interactive proof systems are constructed in two models**
- Random Oracle (RO) Model
  - Parties have access to an RO [BG93, Mic94]

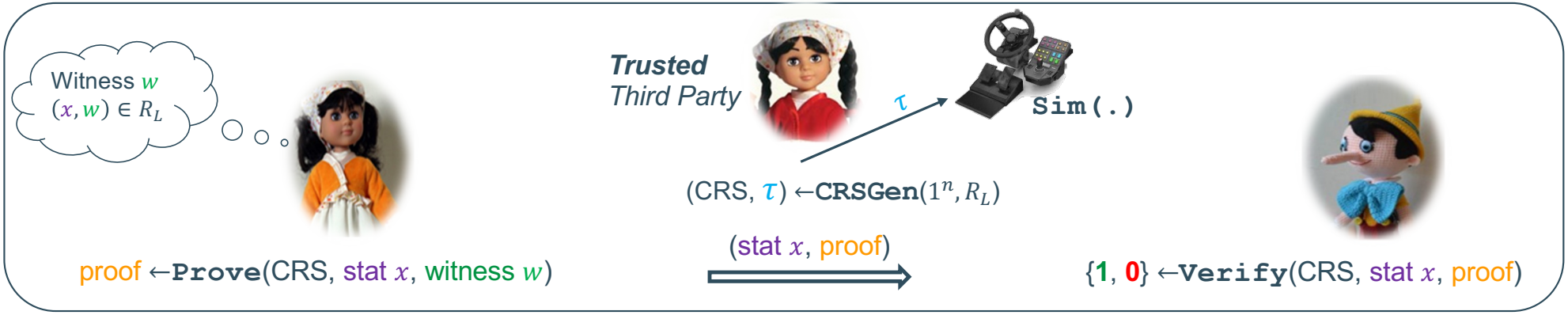


  - In practice, RO is instantiated with hash functions

- Common Reference String (CRS) Model
  - Trusted Third Party generates an SRS [BFM88]



NIZKs in the CRS Model: Completeness, Know. SND & ZK



- **Completeness:** honest P always will convince the honest V.
- **Knowledge Soundness (KS):** dishonest P cannot convince honest V, unless he knows some secret "witness"
- **Zero-Knowledge (ZK):** dishonest V learns nothing more than the truth of the statement.
  - *Formally:* If (CRS, τ) ←CRSGen(1^n, R_L),

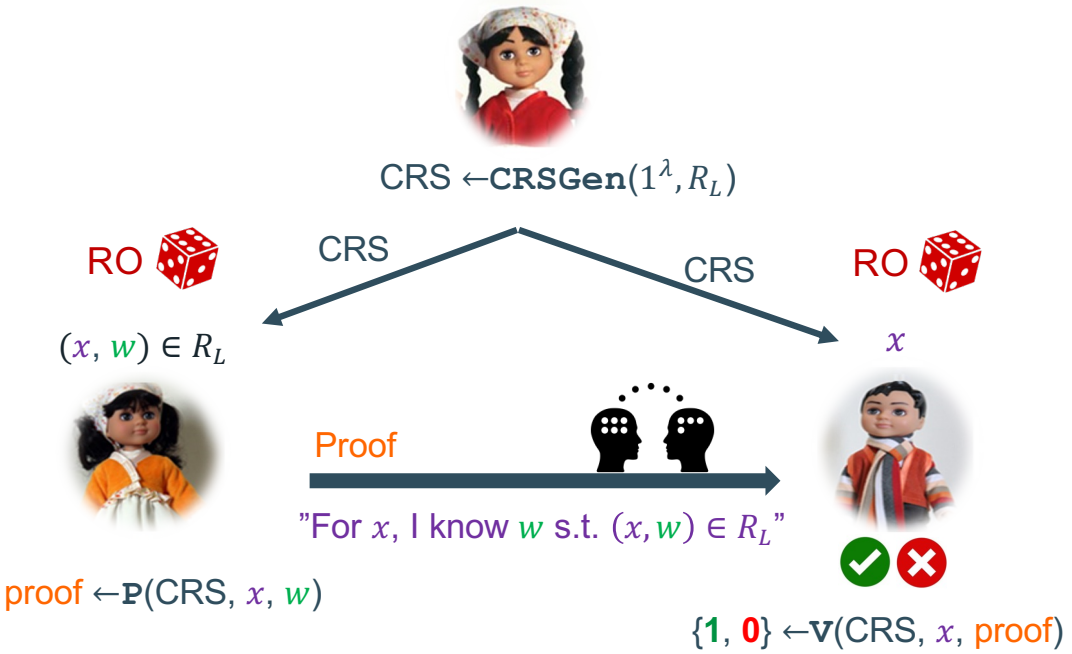
    Pr[proof ← Prove(CRS, stat x, witness w): Verify(CRS, stat x, proof) = 1] =

    Pr[proof ← Sim(CRS, stat x, τ): Verify(CRS, stat x, proof) = 1]

NIZKs in the CRS (or CRS and RO) Model: Constructions

- **CRS:** Can be uniformly random or specific distribution
  - e.g., CRS := 10011101010101011
  - or non-universal CRS := $\left\{\left(g_1^{x^i}, g_2^{x^i}\right)_{i=1}^{d}, g_1^{\frac{\alpha u(x)+\beta v(x)+w(x)}{\gamma}}\right\}$
  - or universal (SRS or) CRS := (g, g^x) or $\left(g_1^{x^i}, g_2^{x^i}\right)_{i=1}^{d}$
- **CRS Generation:** Can be
  - done by a Trusted Third Party (TTP) [Gro16, …]
  - done by MPC protocols [BCG+14, BGM17, ABL+19]
  - in multi-string model (where majority of strings are honest) [GO07]
  - done by a Verifier [BFS16, ABLZ17, Bag19]
  - updatable (MPC with dynamic parties) [GKM+18, ARS20, BS21, BB22]
  - Trusted hardware (e.g., Intel SGX)
  - …

- Common Reference String (CRS) Model
  - A trusted CRS is shared among the parties [BFM88]



CRS of Groth16 [Gro16] zk-SNARK: Non-Universal

- **CRS of Groth16 can be divided into two parts**
  - Non universal elements
  - Universal elements
    - can be generated by an MPC protocol with dynamic parties
    - These types of CRS/SRS are called updatable

Notation:
- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$: bilinear groups
- $[.]_1, [.]_2$: elements from $\mathbb{G}_1$ and $\mathbb{G}_2$
- n : number of mul. gates (or constraints)
- ℓ : number of input wires
- m : number of (input + intermediate) wires

Trusted CRS Generator

$\sigma \leftarrow \mathtt{Setup}(1^\lambda, R_L)$

A Non-Universal CRS σ

$(\sigma, \tau) \leftarrow \mathsf{Setup}(R)$: Pick $\alpha, \beta, \gamma, \delta, x \leftarrow \mathbb{Z}_p^*$. Define $\tau = (\alpha, \beta, \gamma, \delta, x)$ and compute $\sigma = ([\boldsymbol{\sigma}_1]_1, [\boldsymbol{\sigma}_2]_2)$, where

$$\boldsymbol{\sigma}_1 = \left(\begin{array}{c} \alpha, \beta, \delta, \{x^i\}_{i=0}^{n-1}, \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma} \right\}_{i=0}^{\ell} \\ \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\delta} \right\}_{i=\ell+1}^{m}, \left\{ \frac{x^i t(x)}{\delta} \right\}_{i=0}^{n-2} \end{array} \right) \qquad \boldsymbol{\sigma}_2 = \left(\beta, \gamma, \delta, \{x^i\}_{i=0}^{n-1} \right).$$

Powers of Tau (PoT) Protocol:

Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model

Sean Bowe sean@z.cash Ariel Gabizon ariel@z.cash
Ian Miers imiers@cs.jhu.edu

May 3, 2019

ia.cr/2017/1050

BGM17 MPC Protocol: CRS Generating for Groth16

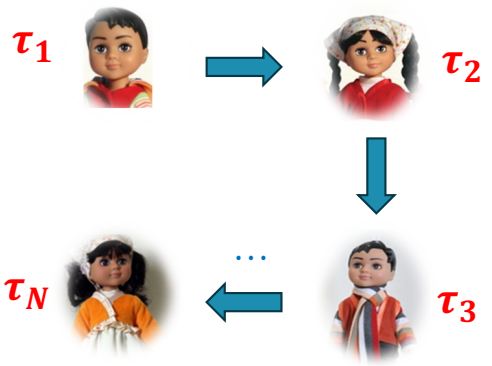
❑ **A two-round MPC protocol**

- Round 1: "*Powers of* τ" a round-robin MPC protocol for generating the universal elements.
- Phase 2: An MPC protocol with fixed number of parties to generate the non-universal elements.

❑ The goal is for parties to contribute secrets to the universal SRS s. t. no single party knows the final secret τ.

- ✓ Given $[\tau]_1, [\tau^2]_1, [\tau^3]_1, \dots, [\tau^k]_1$
- ✓ Party i sample personal share τ_i
- ✓ Computes $[\tau\tau_i]_1, [\tau^2\tau_i^2]_1, \dots, [\tau^k\tau_i^k]_1$

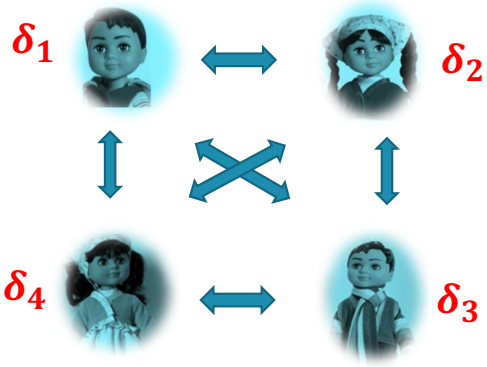
Round 1: "Powers of τ"



They compute the Universal CRS M_1

$$M_1 = \left\{ \begin{matrix} \{[x^i]\}_{i\in[0..n-1]}, \{[x^i]_1\}_{i\in[n..2n-2]}, \\ \{[\alpha x^i]_1\}_{i\in[0..n-1]}, [\beta], [\delta]\{[\beta x^i]_1\}_{i\in[1..n-1]} \end{matrix} \right\}$$

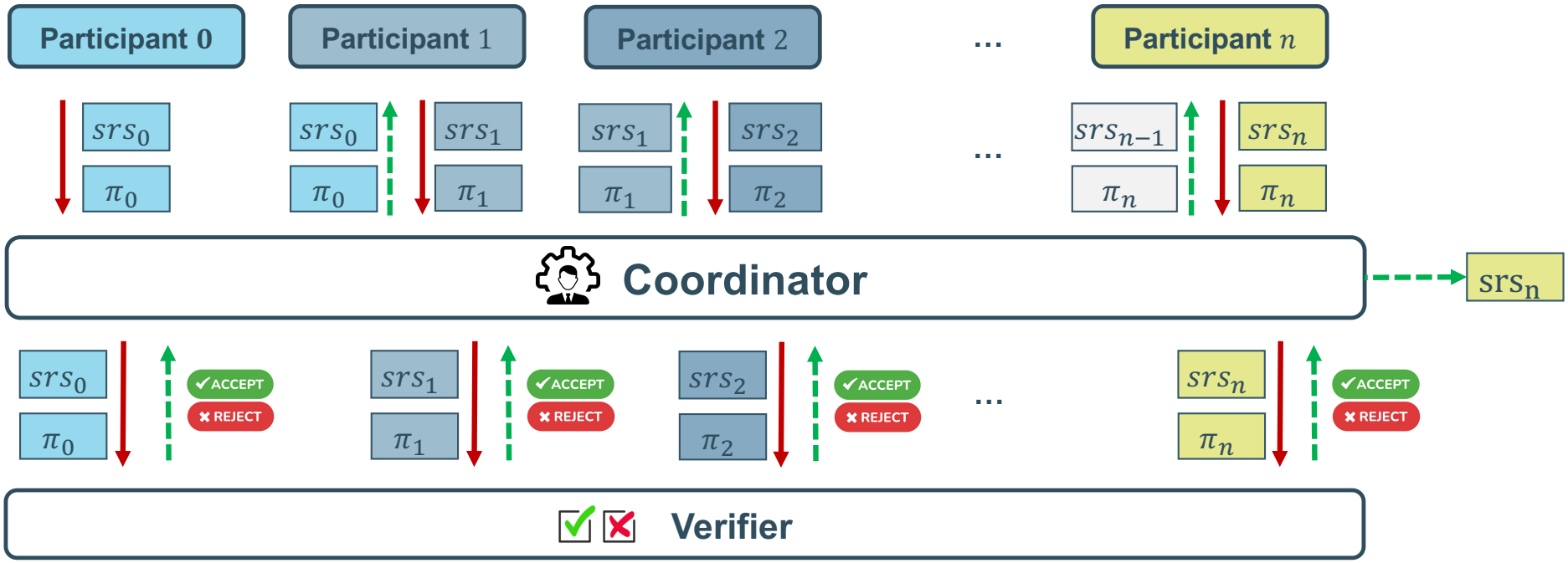
Round 2: Non-universal CRS



A fixed set of parties (not necessary the same ones from Round 1) compute the rest of CRS elements that are non-universal.

The Powers of Tau Protocol: Graphical Representation

❑ **The Powers of Tau Protocol Uses a Coordinator**



Performance of the PoT Protocol: Prover and Verifier

- **Proof generation and verification in Powers of Tau [BGM17]**
  - For Groth16, proving times take less than 16 minutes for all circuit of up to 2^{21} multiplication gates.
  - Verification for each participant takes less then 55 minutes.

"We stress that verification is not run by individual users, it is done by the coordinator and anyone who wishes to verify the transcript of the protocol after completion."
[BMW17]

"Individual participants need not run the verification function."
[BMW17]

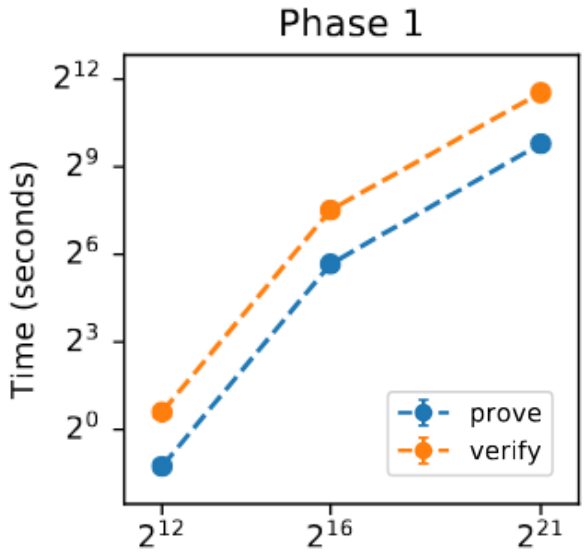
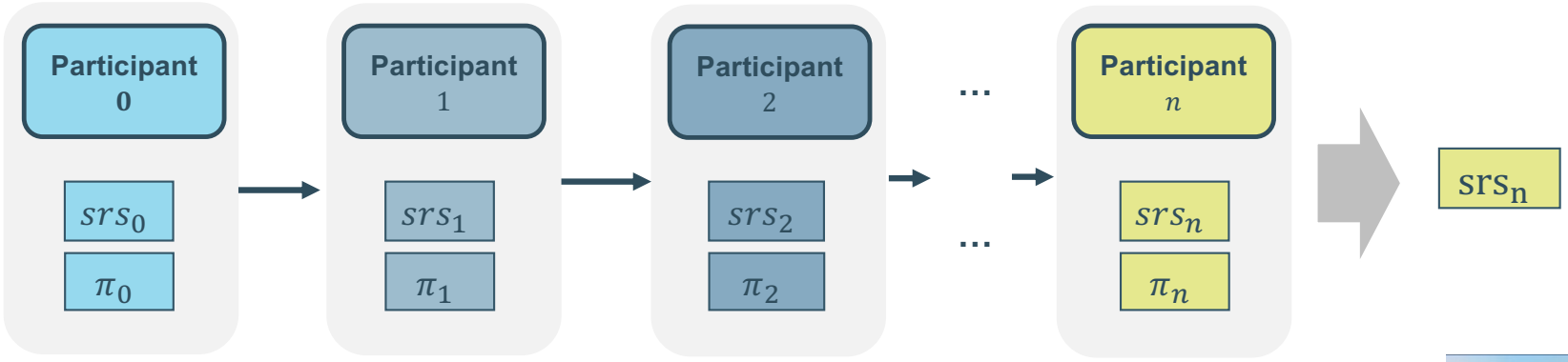


Figure 7.1: Performance of MMORPG protocol phases. Averages taken over 5 iteration. Costs for phase 1 and 2 given for both prove and verification time. Individual participants need not run the verification function. Proving times take less than 16 minutes for all circuit sizes. Verification takes less then 55 minutes. We stress that verification is not run by individual users, it is done by the coordinator and anyone who wishes to verify the transcript of the protocol after completion.

The Powers of Tau Protocol: With Honest Coordinator

❑ **CRS generation ceremony with an honest coordinator acts as a round-robin MPC protocol**



The coordinator is expected (trusted!) to reliably forward messages and verify all proofs! 😇✨

But what happens if the coordinator cannot be trusted!? 🤔🧐

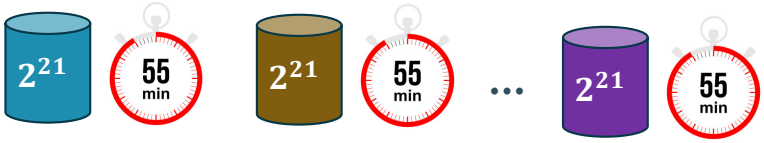
Verification of Original Powers of Tau is Impractical:

~ 55 min

to verify a single update

for a circuit of 2^{21} multiplication gates (Groth16-scale)

Multiply by hundreds of contributors for every end user.

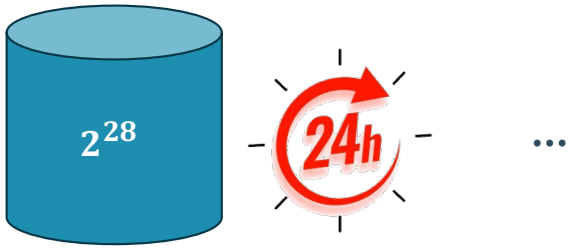


Why so slow?

- Each SRS element must be checked for consistency.
- The check is a **pairing equation**
- A ceremony of size 2^{21} means millions of pairings.

❑ **Some ceremonies have sample SRS of size 2^{27} or 2^{28}**

- Significantly slower verification!
- E.g., In "Perpetual Powers of Tau" ceremony,
  - By the semaphore team; CRS size: 2^{28} , Number of participant: 71
  - Each contribution required a 97 GB download, 1-day computation, and a 49 GB upload [Fou21].



Receipts from the Perpetual PoT Ceremony: Verification

Perpetual Powers of Tau Ceremony — Google Group · Dec 5, 2019

Subject: Transcript verification up to challenge #15

> *I would like to express my thanks to Brice Huang and Brian Gu for their excellent work on powersoftau-verifier, with which we have successfully verified the transcript up to the current challenge file (#15).*

> *The results of the verification, which took **1.5 weeks** on an Azure Standard F4s_v2 (4 vcpus, 8 GiB memory) VM, can be found here.*

— Koh Wei Jie

Perpetual Powers of Tau Ceremony coordinator

1.5 weeks

of compute

to verify the first 15 updates of Perpetual Powers of Tau

For 71-participant it would take over 7 weeks!!!

What this tells us

- No end-user will ever run the original verifier; the cost is simply out of reach.
- Even a handful of updates is a multi-week job on a dedicated VM.
- In practice, only the coordinator checks them!

https://groups.google.com/a/ethereum.org/g/perpetual-powers-of-tau-ceremony-group/c/7WiPMo7GbIQ

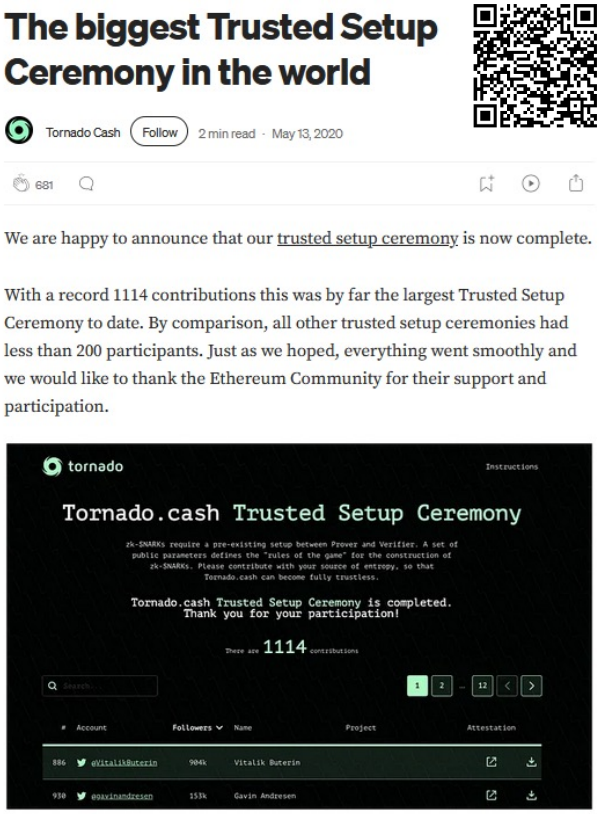
Hidden Trust and Impractical Verification in the PoT:

- Seems the statements like

"Individual participants need not run the verification function."
[BMW17]

- And the extremely slow verification time, has likely been the main reason for conducting ceremonies with many participants without explicitly asking all the participants or each individual end-user to verify the final SRS!

⚠ **Tornado.cash ceremony have had 1114 participants!**



The biggest Trusted Setup Ceremony in the world

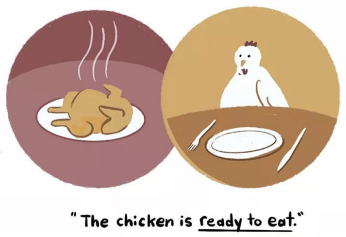
Tornado Cash Follow 2 min read · May 13, 2020

681

We are happy to announce that our trusted setup ceremony is now complete.

With a record 1114 contributions this was by far the largest Trusted Setup Ceremony to date. By comparison, all other trusted setup ceremonies had less than 200 participants. Just as we hoped, everything went smoothly and we would like to thank the Ethereum Community for their support and participation.

The Story the Ceremony Pages Tell:



"Only one participant needs to do this successfully to ensure the final parameters are secure"

- by Zcash Research Team

"If at least 1 participant is honest and destroys their secret, then there should be no way for a malicious prover to create fake proofs"

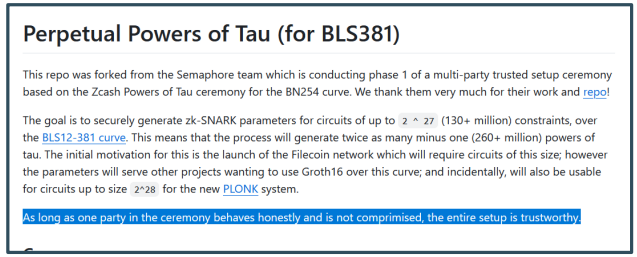
- on Aleo's Git Page

"As long as one party in the ceremony behaves honestly and is not comprimised, the entire setup is trustworthy"

- on Ethereum Foundation Blog
- by File Coin Research Team
- on Loopring's Git Page

➢ **and many similar ones, made on different web pages are not clear enough …**

Seems the BGM framing has propagated into the documentation of real ceremonies. All statements emphasize ***"1-of-n honest"***
— **non mention who verifies, or what the end user is supposed to check.**


Perpetual Powers of Tau (for BLS381)

This repo was forked from the Semaphore team which is conducting phase 1 of a multi-party trusted setup ceremony based on the Zcash Powers of Tau ceremony for the BN254 curve. We thank them very much for their work and repo!

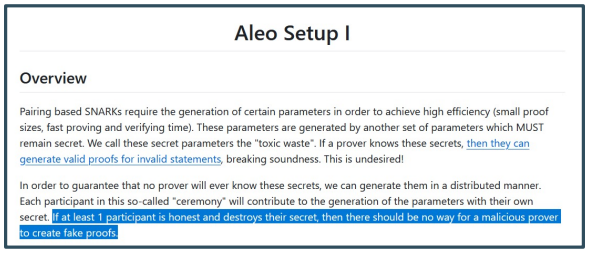
The goal is to securely generate zk-SNARK parameters for circuits of up to 2 ^ 27 (130+ million) constraints, over the BLS12-381 curve. This means that the process will generate twice as many minus one (260+ million) powers of tau. The initial motivation for this is the launch of the Filecoin network which will require circuits of this size; however the parameters will serve other projects wanting to use Groth16 over this curve; and incidentally, will also be usable for circuits up to size 2^28 for the new PLONK system.

As long as one party in the ceremony behaves honestly and is not comprimised, the entire setup is trustworthy.


Loopring ZKP Trusted Setup Multi-party Computation Ceremony

We use a very similar method of contributing as in phase 1 of the ceremony. As long as one party in the ceremony behaves honestly and is not comprimised, the entire setup is trustworthy.

To monitor our ceremony's progress, please visit https://loopring.org/#/ceremony.


Aleo Setup I

Overview

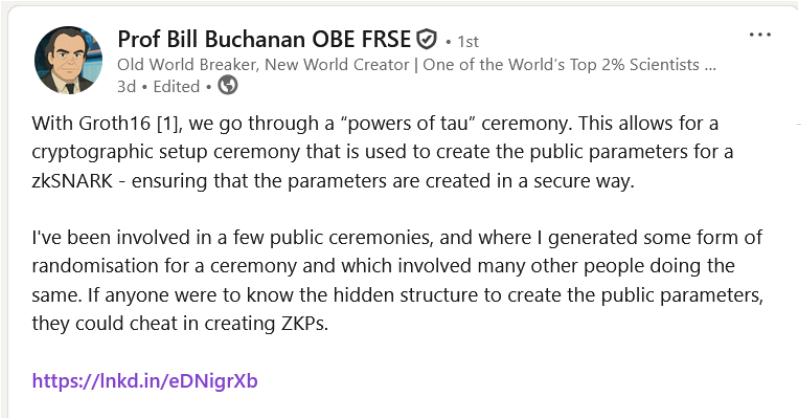
Pairing based SNARKs require the generation of certain parameters in order to achieve high efficiency (small proof sizes, fast proving and verifying time). These parameters are generated by another set of parameters which MUST remain secret. We call these secret parameters the "toxic waste". If a prover knows these secrets, then they can generate valid proofs for invalid statements, breaking soundness. This is undesired!

In order to guarantee that no prover will ever know these secrets, we can generate them in a distributed manner. Each participant in this so-called "ceremony" will contribute to the generation of the parameters with their own secret. If at least 1 participant is honest and destroys their secret, then there should be no way for a malicious prover to create fake proofs.

Many Participants Have Not Verified The Final CRS:

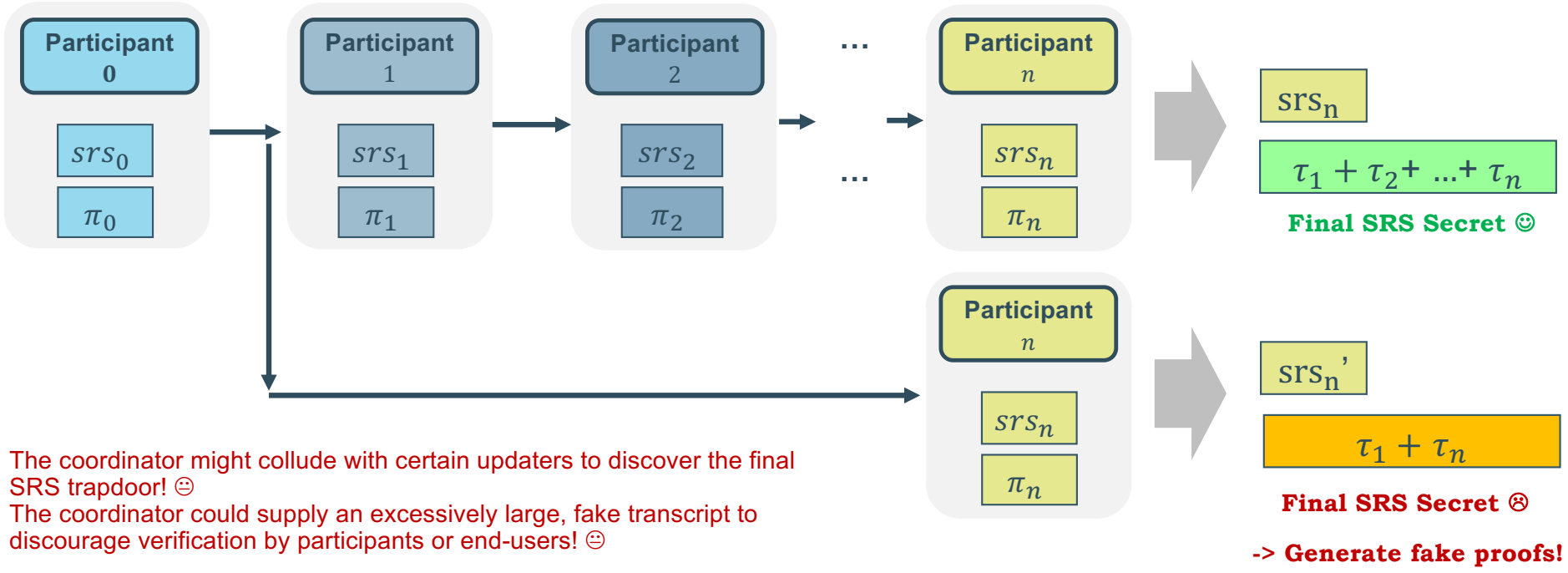
- ❑ **It is crucial to ensure that end-users are explicitly informed of their options.**
- ❑ **While some ceremonies have implemented and provided the original SRS verification algorithm of the Powers of Tau protocol, along with the transcript of the ceremony, the slow nature of their SRS verification algorithm renders it impractical.**

- ✓ **It is highly likely that many participants have not verified the final SRS despite the availability of these resources in case of some ceremonies.**



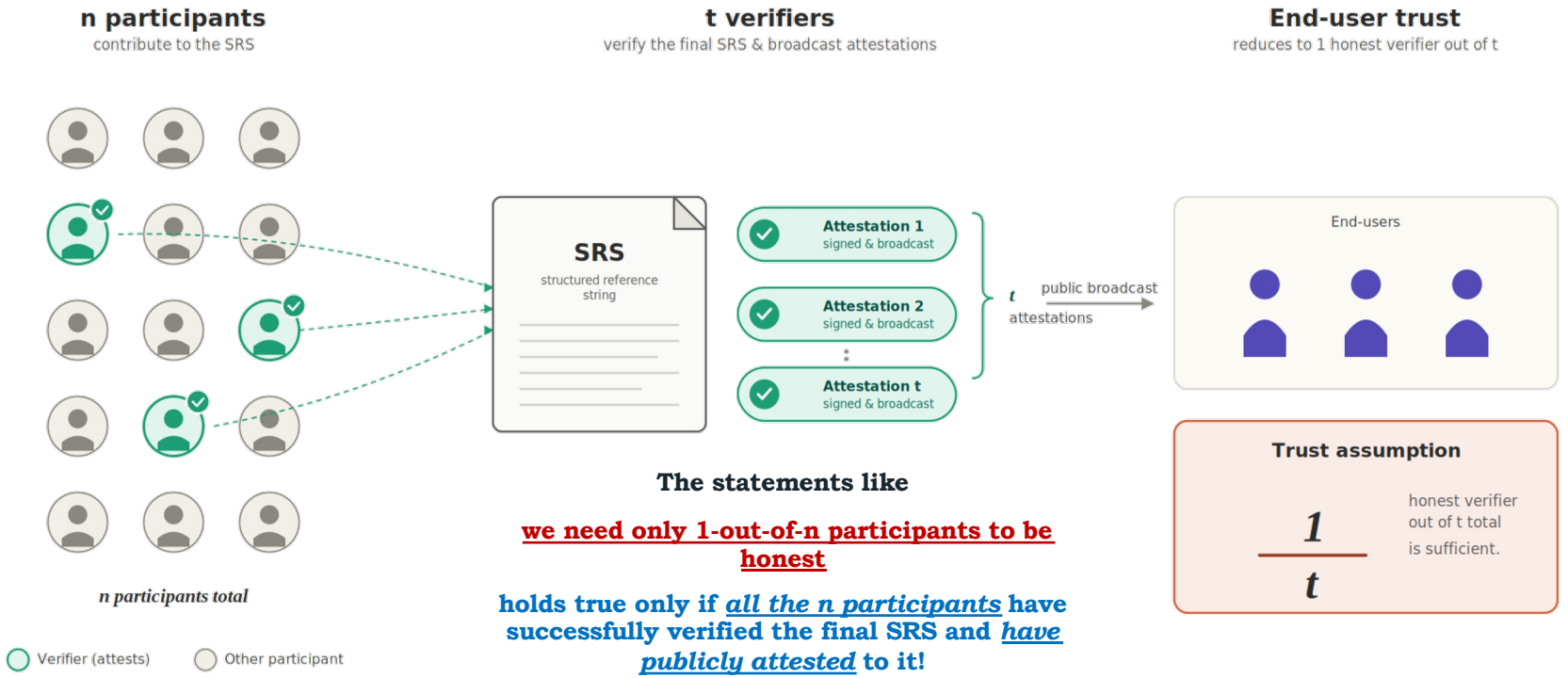
Undetectable Forks by a Malicious Coordinator:

❑ **A coordinator can create a fork, manipulate final SRS and even learn the final SRS trapdoor!**



- ✓ The coordinator might collude with certain updaters to discover the final SRS trapdoor! 😐
- ✓ The coordinator could supply an excessively large, fake transcript to discourage verification by participants or end-users! 😐
- ✓ …

We Show That:



The statements like

we need only 1-out-of-n participants to be honest

holds true only if *all the n participants* have successfully verified the final SRS and *have publicly attested* to it!

If t out of n participants verify the final SRS and broadcast attestations, end-users only need to trust 1 honest verifier among the t.

Solutions:

- **Naïve Solution:** Extra to One of the Participants, Trust the Coordinator!

One of the Participants + **Coordinator (Much Bigger Trust!)**

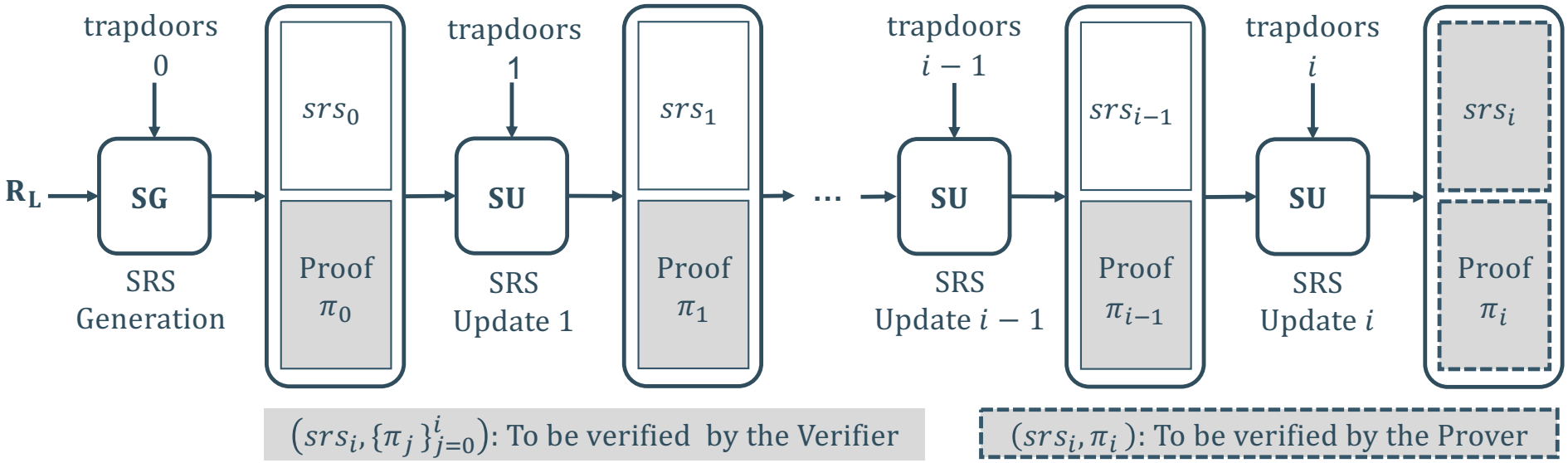
⚠ Contradicts the goal of distributed SRS generation, which aims to distribute trust among multiple parties!

- **Our Proposed Solution:** Verify Chain of Proofs + The Final SRS
  - Either all ceremony participants individually should verify the final SRS and securely broadcast their attestations for successful verification.
  - Or each end-user must verify the final SRS, using the original transcript of the protocol (signed by the participants).

⚠ **In both cases still the end-user needs to trust one of the participants (to destroy their share)!**

Aggregated and Batched Verification:

❑ **We propose a practical verification algorithm for the PoT Protocol:**



❑ **Our final proposed verification algorithm uses both the aggregation and batched techniques.**

❑ **The algorithm relies on a corollary of the Schwartz-Zippel lemma.**

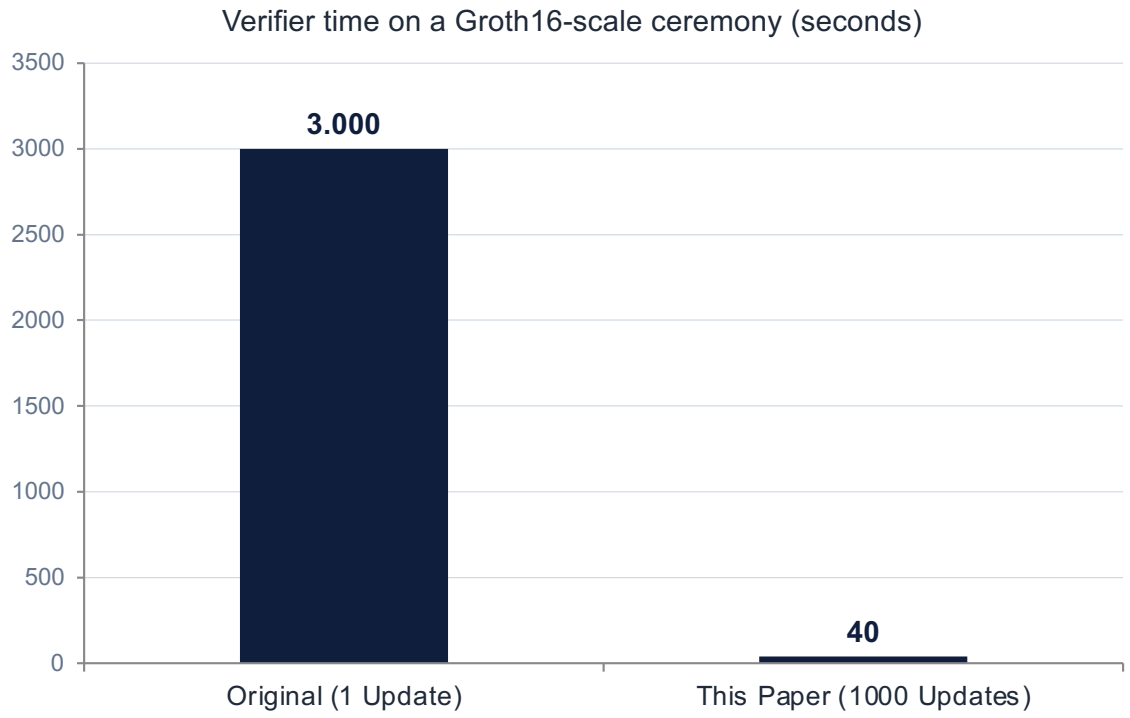
Identifying a Malicious SRS Updater: Naïve & Recursive

- To identify a malicious updater one needs to run the (B)SV algorithm after each update.
- We propose a more efficient approach based on recursive execution of BSV algorithm [BMS23].

Performance and Concrete Impact:

Transparency

Efficiency



The aggregated and batched check replaces hours of pairing work with a few seconds of exponentiations + a constant pairing check.

What changes in practice

- End-users can verify in-browser.
- Pair-based wallets & rollups no longer need to trust a coordinator.
- Ceremony failures become debuggable — blame in log n rounds.
- Drop-in for existing Powers-of-Tau transcripts — no re-run.

✓ **New algorithms can be adapted for a specific ceremony.**

KU LEUVEN

Thank You!

ia.cr/2025/2000

✉ karim.baghery@kuleuven.be