**Simplifying Powers of Tau and the Trusted Setup Ceremony.**

[[](https://medium.com/@iamagnix?source=post_page-----c9a4f1833dc2--------------------------------)](https://medium.com/@iamagnix?source=post_page-----c9a4f1833dc2--------------------------------)

[[Coinmonks](https://medium.com/coinmonks?source=post_page-----c9a4f1833dc2--------------------------------)](https://medium.com/coinmonks?source=post_page-----c9a4f1833dc2--------------------------------)

[Agnish Ghosh.](https://medium.com/@iamagnix?source=post_page-----c9a4f1833dc2--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F4998122a7661&operation=register&redirect=https%3A%2F%2Fmedium.com%2Fcoinmonks%2Fsimplifying-powers-of-tau-and-the-trusted-setup-ceremony-c9a4f1833dc2&user=Agnish+Ghosh.&userId=4998122a7661&source=post_page-4998122a7661----c9a4f1833dc2---------------------post_header-----------)

Published in

[Coinmonks](https://medium.com/coinmonks?source=post_page-----c9a4f1833dc2--------------------------------)

·

6 min read

·

Apr 30

25



before any reax at all, this pic is by DALL-E 2.

Prompt: husband putting a ring with an Ethereum logo on it, in wife’s finger

In this blog, I’m making an attempt to break down Powers of Tau. One, indeed confusing part while writing zkSNARK circuits, be it in Circom, or Rust, or any other language, that many developers usually struggle with.

Powers of Tau is a ceremony that is used to generate the initial parameters for zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge). The ceremony is used to ensure the security of the initial parameters by using a multi-party computation (MPC) protocol to generate them. The main idea behind Powers of Tau is that by having multiple participants contribute randomness to the generation of the parameters, it becomes much more difficult for any individual machine/node, or group to tamper with them or compromise the entire security of the system.

Starting with breaking down what a zkSNARK is….

Zero-Knowledge Succinct Non-Interactive Argument of Knowledge.

The term “zero-knowledge” refers to the fact that the proof reveals no information beyond the truth of the statement being proved. The term “succinct” refers to the fact that the proof is much shorter than the computation that it attests to, making it efficient to transmit and verify. Finally, the term “non-interactive” refers to the fact that the proof can be generated and verified without any interaction between the prover and verifier beyond the initial exchange of information.

zkSNARKs are widely used in various applications, including privacy-preserving transactions in cryptocurrencies like Zcash, authentication protocols, and secure computation in decentralised applications.

They offer a powerful tool for enabling secure and private computation without the need for trust in any single party.

Now what are the parameters of a zkSNARK system?

In a zkSNARK, there are several parameters that are used to define the proving system and ensure its security. The main parameters include:

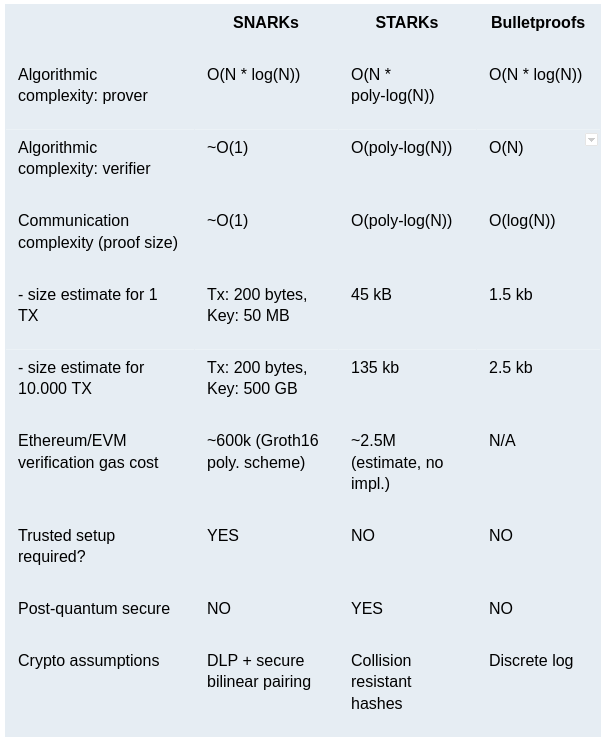
1. Public Input: This is the information that is publicly know and can be accessed by anyone. It’s usually the input to the computation that the proof is being generated for.
2. Secret Input: This is the information that is kept secret and is used to generate the proof.
3. Verification Key: This is a public key that is used to verify the correctness of the proof. It’s generated from the secret inputs and is made public in order to be used to verify the proofs.
4. Proof: This is a succinct representation of the computational constraints that are being proved. It’s generated from the secret input and can be verified using the verification key.
5. Proving Key: This is a secret key that is used to generate proofs. It’s generated from the secret input and must be kept private to ensure security of the proving system.
6. Common Reference String (CRS): This is a common set of parameters that are used to generate the proving and verification keys. The CRS is generated using the MPC protocol that we’re talking about here, Powers of Tau.

With the advent of STARKs (Zero Knowledge Scalable Transparent Arguments of Knowledge), trusted ceremonies are at times getting redundant, because STARKs essentially don’t need one. Thus, sometimes CRS is also sometimes called “Toxic Waste”, no I didn’t call you one. 🙂

Note that the Powers of Tau ceremony is ONLY required in Non-Interactive Zero-Knowledge Proofs, more specifically SNARKs, that’s because STARKs don’t need a trusted setup ([Halo2](https://zcash.github.io/halo2/index.html), Polygon [Miden](https://polygon.technology/polygon-miden" \t "_blank), [Starknet](https://docs.starknet.io/documentation/" \t "_blank), etc).

Also [Bulletproofs](https://eprint.iacr.org/2017/1066.pdf) are another notable ZKP proving system that do not require a trusted setup.

Here’s a comparative study of the most well-known types of ZKP systems.



In Non-Interactive Proofs the interaction between the Prover and the Verifier can be simulated by the prover itself, making the direct communication with the verifier kind of unnecessary (like Interactive ZKPs)

One very common application of Powers of Tau today is Miners in a Rollup facilitating blockchain network. You can read more about it here.

Until now, if you think how big of a deal is randomness, I would really suggest you to watch this video, where [this man got a piece of fabric from Chernobyl Exclusion Zone, just to get the measure for radioactivity onto a Geiger Counter, so that he could randomise that.](https://www.youtube.com/watch?v=I4cDAqeEmpU)

Why do we call it “Powers of Tau” though?

That is because we use the powers of the respective generator functions of the numbers within a group, belonging to each actor/participant in Multi-Party Computation, the final parameter set we’re generating using this is denoted as Tau.

And, the main reason why we call them Trusted Setup Ceremonies is indeed complicated.

Even if you look up in the aforementioned table, zkSNARKs essentially beat Bulletproofs and STARKs in 2 places, proof size, and the verification time. From the product perspective, let’s I am writing a zkSNARK for zk identity, the proof can be a one-time thing and can be stored in the browser, offline in a file. What essentially should be fast is time for identity verification, or we can say the authentication, which in turn depends the most on the proof size and the standard verification time.

Here, the verification side can exist either as a Solidity smart contract deployed on the corresponding EVM compatible chain, or simply an externally hosted web server. More importantly, so far to build an application specific reusable zkp proving system, zkSNARks are pretty much the only option in the industry.

Now, the biggest hurdle left to cross is the degree of security of the CRS. In a web2 space the generation of a CRS would definitely be easy, by included internalised private member machines, but that defeats the purpose of decentralisation entirely. To facilitate the CRS generation process in a web3 space, we are left with Multi-Party Computation.

The primary job of MPC schemes is check whether no single entity generates the entire CRS, or in other words, is able to gain entire knowledge of the underlying math of the CRS. This is achieved by allowing more and more parties to participate in the MPC scheme, but in such a way that we only need a minority of the participants to be honest. The more honest participants we find, the more randomness is added onto the CRS generation (no need of Chernobyl Fabric anymore), the more secure the system gets.

There are several variants of this same generation process, but usually it goes through several rounds of participants, sometimes serially, sometimes concurrently depending upon variant to variant, and sometimes is substantially time-taking.

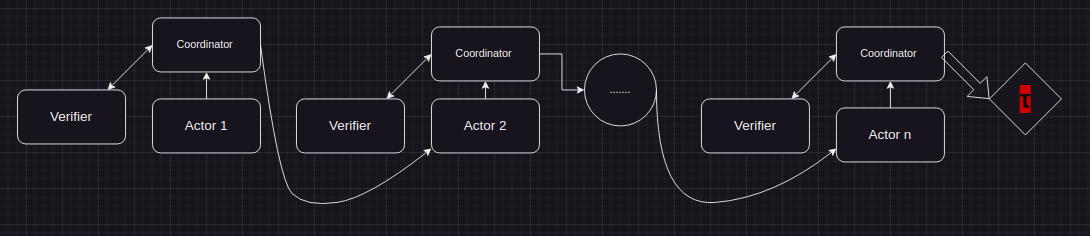
Because of this tedious usage of multi-actor gathering and multiple rounds of participation, we call this process Trusted Setup Ceremony.

One notable variant of Powers of Tau was mentioned in [MMORPG](https://eprint.iacr.org/2017/1050.pdf) paper recognized by the [International Association for Cryptologic Research.](https://iacr.org/)

This Trusted Setup Ceremony included (1 + n) rounds where n is the number of actors active for MPC. The actors included random actors and coordinator, where coordinators always happen to be the honest one’s, verifying other actor’s job in giving randomness.

The first round is spent mainly on the initialisation of the mathematical circuits for Powers of Tau. The second round onwards the actors started participating.

The primary drawback of this system was that actor participation for the MPC happened in a serial order. Which meant that the participants could participate one at a time. As the constraints in the zkSNARKs kept on increasing linearly, the CRS to be generated grew with the same linearity, thereby, needing more and more paritcipants in a serial order, which was in fact time-taking, hence, stop attracting users at a later stage.



MMORPG

Later on, Powers of Tau underwent several improvements which facilitated concurrent participation of actors in the Trusted Setup ceremony.

<https://medium.com/coinmonks/simplifying-powers-of-tau-and-the-trusted-setup-ceremony-c9a4f1833dc2>