Week 1

Part 1

Two types of SNARK proofs

- PLONK
- GROTH16

Why SNARK requires a trusted setup while STARK doesn’t

SNARKs are based on elliptic curve cryptography, similar to public key encryption. The SNARK setup produces toxic waste (the private key) from which the public key could be rederived, which could be used to simulate a proof. STARKs are based on a different primitive (hash collision resistance) where no such toxic waste is produced.

Name two more differences between SNARK and STARK proofs

STARKs are quantum resistant, but they don't have a constant proof size.

Part 2

What does the circuit in `HelloWorld.circom` do?

It checks that two private inputs multiply to a public output.

What is a Powers of Tau ceremony?

Powers of Tau is a way of generating a trusted setup where many parties concatenate their trusted computations. The entire cerenomy is secure as long as there is just a single honest participant, so increasing the number of participants increases the security of the protocol and the result can be reused by any future project, thereby increasing scalability (there is still a second part to the ceremony that is project-specific).

How are Phase 1 and Phase 2 trusted setup ceremonies different from each other?

As mentioned above, the second phase of the setup is specific to the circuit, so to the problem that the project is tackling, and thus can't be reused by all projects unlike the phase 1 output discussed above.

Try to run `compile-Multiplier3-groth16.sh`. You should encounter an error with the circuit as is. Explain what the error means and how it arises.

Circom only allows quadratic constraints, ie only two solutions are allowed to be multiplied. To multiply three variables, create a temporary variable.

How is the process of compiling with PLONK different from compiling with Groth16?

PLONK does not require a second phase.

What are the practical differences between Groth16 and PLONK?