Trusted Setup & Sonic Performance Report

Decentriq AG

This report gives an overview of the MPC ceremony carried out for Zcash' Sapling release and discusses the applicability of the ceremony to the Filecoin circuits using **Groth16**, as the underlying proof system. In addition to this, we provide performance metrics for the computation of the Blake2s hash function using **Sonic**, both in helped and unhelped mode, as well as a short discussion on the applicability of **GM17** to Filecoin proofs. We conclude by providing an outlook on the feasibility of a trusted setup ceremony for Filecoin.

Trusted Setup

The goal of the trusted setup is to generate a common (or structured) reference string (CRS or SRS) for proving and verification of *succinct* arguments of knowledge. The first commercial ceremony for generating such parameters was carried out by Zcash for their **Sprout release**. The ceremony used an MPC procedure, which required all participants to be online throughout the duration of the computation, and thus severely limited the number of parties that could join the ceremony. To increase the number of contributors and reduce the chances of attacks or collusion, a **subsequent ceremony** was carried out for the Sapling release, which utilized a new MPC scheme, called Powers of Tau, that alleviated the requirement of participants being simultaneously online.

Powers of Tau

Powers of Tau is a two-stage MPC protocol, which allows participants to contribute their shares towards the CRS *sequentially* and allows them to verify that their contribution is present in the final parameters. The ceremony relies on an untrusted centralized coordinator, which communicates a "challenge" to each participant, receives their "answer" and forwards a challenge to the next participant based on all "answers" received up until this point.

The basic intuition behind the procedure is that each person contributes some randomness to the parameters and that the final parameters are produced by combining the randomness from all participants. To simplify, a list such as (g, g, … , g) is sent by the coordinator to the first participant, where g is a generator point of a finite cyclic group of prime order. The participant chooses a random "a" and computes ($a * g$, $a^2 * g$, …, $a^n * g$) and makes the data publicly available. The next participant takes the output, choose a random "b" and computes ($b * a * g$, $b^2 * a^2 * g$, …, $b^n * a^n * g$) and also makes it publicly available. This procedure is repeated for all remaining participants. In the end, a random beacon is applied, to avoid any potential adaptive attacks from the last contributor.

Some of the advantages of the Powers of Tau over the Sprout ceremony are:

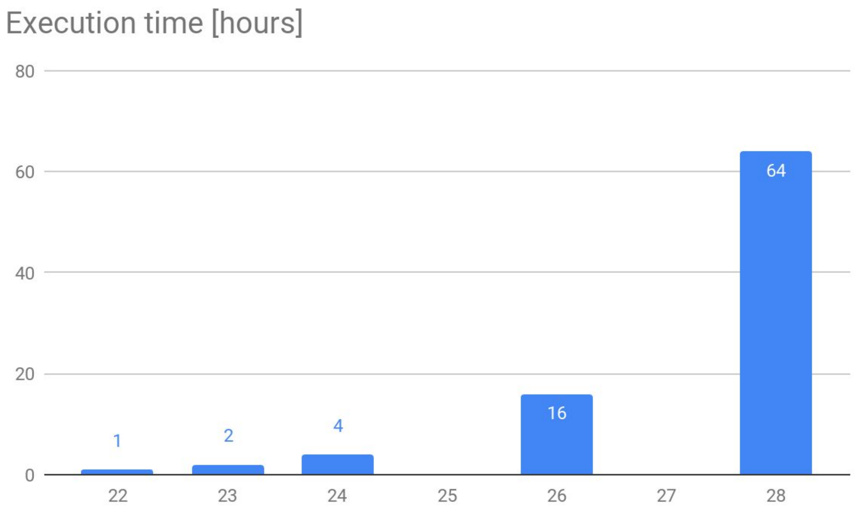
- There is no pre-commitment round, which means participants are not fixed; instead to avoid adaptive adversaries a random beacon is applied at the end of the procedure (e.g. 2^{42} SHA-256 iterations on the Bitcoin nonce in a given block, as done in Sapling)
- The first round of the Powers of Tau produces generic parameters for all circuits up to a given size
- The second round, which produces circuit-specific parameters, is not computationally intensive
- The second round does not require the same set of participants as the first round and can produce a smaller parameter set if suitable for the application

Performance

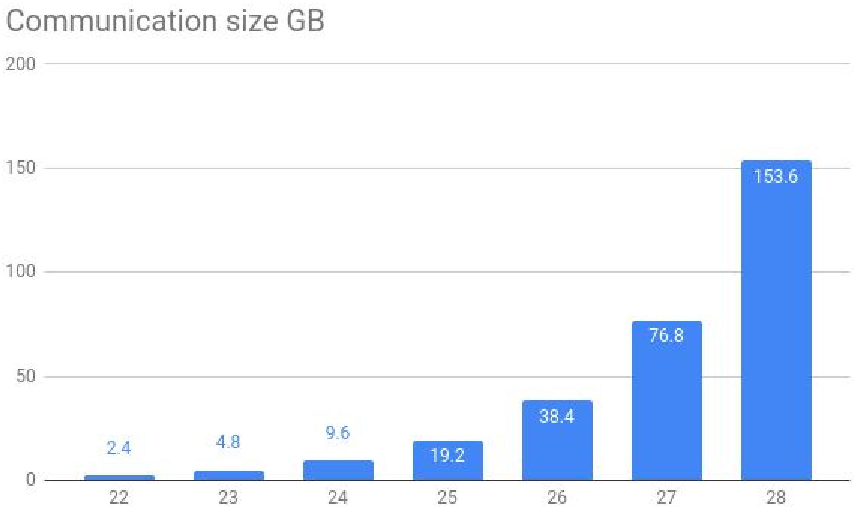
We re-ran the two phases of the Powers of Tau ceremony, using the available implementation of **Phase 1** and **Phase 2** of the protocol for the Sapling release (the code generates the CRS for Groth16). The test runs were done on a Linux machine, running Ubuntu 16.04, with 32 GB RAM and 6 Intel i7-8700K cores @ 3.7 GHz. The output of Part 1 provides generic parameters for all circuits with upto 2^{21} constraints, while the output of Phase 2 produces the Sapling circuit-specific parameters. The running time for the first part was 1h 29 minutes and the memory consumption was 2 GB, while for the second part it was 15min 45 seconds and the memory consumption was 1.5 GB.

We note that the running time of both phases increases linearly with the number of parameters, and the same holds for the communication overhead (the amount of data that needs to be transferred between participants). For Phase 1 of the Zcash ceremony, each participant needs to download a 1.2 GB challenge file and to upload a 577 MB response file. For Phase 2 of the Zcash ceremony, each participant needs to download a 742 MB file and upload a file of the same size. This means that in total, each participant needs to download ~2 GB of data and upload around 1.3 GB of data to participate in both phases.

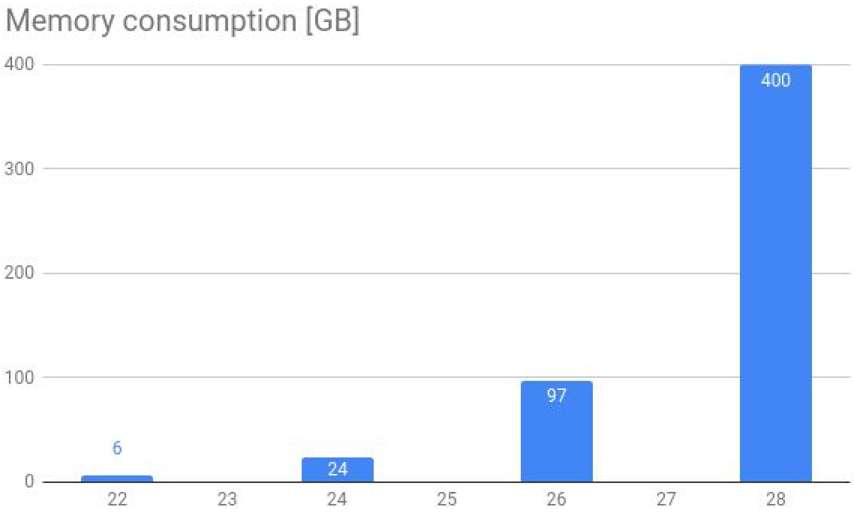
We compared those numbers with estimates provided by Gnosis for running Phase 1 of a trusted setup ceremony on a machine with 15 cores @ 3.2 GHz, depicted in the plots on the next page (where the x-axis is the exponent for the number of constraints, i.e. 2^{22} constraints, 2^{23} constraints, etc.).



Execution time [hours]



Communication size GB



Memory consumption [GB]

We note that the numbers are (mostly) consistent with our test runs, with the following differences:

- ➢ The communication size only reflects the size of the challenge file that needs to be downloaded and doesn't account for the response file, which needs to be uploaded.
- ➢ The execution time, estimated by Gnosis, is slightly faster than the expected execution time based on our test run for 2^{21} constraints, i.e. extrapolating from our results we would expect that it would take ~3 hours to compute the parameters for 2^{22} constraints on 6 cores @ 3.7 GHz, which should translate to more than 1 hour on 15 cores @ 3.2 GHz.
- ➢ The memory consumption we estimate for 2^{22} constraints should be 4 GB instead of 6 GB. This is a substantial difference, however, we note that the implementation of Phase 1 and Phase 2 by Zcash is non-optimal in terms of memory consumption, as it loads all the parameters in memory, instead of streaming them, which means that the memory consumption can be kept constant by an optimal implementation. Hence, we are not concerned with this difference in the report.

Applicability to Filecoin

The Sapling ceremony produced generic parameters for all circuits up to a size of 2^{21} multiplicative constraints. Since the size of the circuits in Filecoin is much larger than 2^{21}, this means that the parameters cannot be reused and the whole ceremony must be repeated from scratch.

Based on our test runs, we arrive at the following performance metrics (per participant) for the trusted setup of a circuit with 879 million constraints (which corresponds to the replication of ⅛ of a 64 GiB sector):

Num cores	Num constraints	Execution time (hours)	Communication size download (GB)	Communication size upload (GB)	Phase
6 @ 3.7 GHz	879 million	630	504	242	1
6 @ 3.7 GHz	879 million	105	311	311	2

The above numbers render any trusted setup ceremony for Groth16 with the existing Powers of Tau protocol infeasible for a circuit of such size both due to the execution time and the size of the data to be communicated between the participants and the coordinator. Note that, while in theory it is possible to speed up the execution by utilizing a much more powerful machine, this comes at the drawback of limiting the number and diversity of the participants in the ceremony, as well as its security in case cloud instances are used.

We believe a realistic upper limit on the number of constraints that can be handled by the protocol, without severely limiting it to participants with very powerful hardware, to be around 16 million:

Num cores	Num constraints	Execution time (hours)	Communication size download (GB)	Communication size upload (GB)	Phase
6 @ 3.7 GHz	16 M	11.5	9	4.4	1
6 @ 3.7 GHz	16 M	1.9	5.6	5.6	2

The above numbers would yield a *lower bound* on the amount of time that a single contributor to the ceremony will need to 13.5 h + communication time necessary for uploading 14.6 GB & downloading 10 GB of data. Based on the **Speedtest Global index**, the median download speed in the Top 25 countries is ~110 Mbps. We use a median upload speed of ~55Mbps, extrapolating from the global data, since this information is not available. This yields a total of 25 minutes for the upload and 17 minutes for the download, for an overall lower bound of slightly over 14 hours/participant.

Sonic performance

As part of our evaluation, we collected performance metrics for Sonic on a Blake2s circuit with varying length of inputs. The goal was to understand the performance of Sonic compared to Groth16 and evaluate its feasibility for Filecoin. The code for the test runs as well as a summary of the results is available at: **https://github.com/stefandeml/bellman_benchmarks**.

Overall, our empirical comparison between Sonic and Groth16 indicates that proving in Sonic is 5-10x slower than Groth16, while consuming ~5x more memory. The largest problem instance that we ran successfully on a server with 32 GB RAM and 6 core Intel Core i5-8400 was for 2 million constraints.

Helped mode

The helped version of Sonic, which is required to make verification succinct, induces an additional overhead (over the 5-10x for proving) for generating the advice and for batching the proofs. The overhead from these two operations scales sublinearly with the size of the batch: in particular for a circuit with a fixed number of constraints, the advice creation time is constant and the computation of the aggregated proofs is sublinear. In our test runs, for a Blake2s circuit with 128K constraints (corresponding to 384 bytes of input) computing the advice took 2.4 seconds for batches of size 3, 5, 10, 20 and 40, while the proof aggregation took 9 seconds for a batch of size 3 and around 69 seconds for a batch of size 40. A more detailed breakdown is available in the last table **here**.

GM17

The protocol described in Groth16 has been identified to have a malleability issue, as detailed in GM17. The malleability allows anyone, who has seen a valid proof, to generate a new one, without knowing a valid witness. This malleability, however, is only relevant for protocols, which rely on zero-knowledge. To the best of our knowledge, the Filecoin proofs rely on public inputs to ensure the integrity of the proofs, and hence our tentative conclusion is that it is not necessary to switch to GM17. We, however, deem it necessary to surface this as a potential issue and recommend it undergo internal clarification to ensure that the security of the overall system is not compromised by proof malleability.

Conclusion

In our view, with the current size of circuits in Filecoin, it will be infeasible to carry out a trusted setup ceremony using Groth16 and the existing Powers of Tau protocol. A feasible limit on the amount of constraints that can be handled by the protocol is ~16 million, which is about 55 times smaller than the circuit size for replication of ⅛ of a 64 GiB sector.

The empirical results from our test runs with Sonic seem to indicate that using this protocol instead of Groth16 will incur significant overhead both in terms of computation time and memory usage.

Based on the above, we believe that to make the computation of the Filecoin proofs feasible and to enable a trusted setup ceremony to be carried out, the most promising path forward is to investigate the usage of considerably smaller circuit sizes and batching of proofs, either in Groth16 or in Sonic.

References

Groth16, **https://eprint.iacr.org/2016/260.pdf**
GM17, **https://eprint.iacr.org/2017/540.pdf**
Sonic, **https://eprint.iacr.org/2019/099.pdf**
Zcash Sprout, **https://z.cash/blog/the-design-of-the-ceremony/**
Powers of Tau Ceremony, **https://www.zfnd.org/blog/powers-of-tau/**
Power of Tau codebase, **https://github.com/ebfull/powersoftau/**
Sapling MPC codebase, **https://github.com/zcash-hackworks/sapling-mpc**
Speedtest global index, **https://www.speedtest.net/global-index**