Benchmarking ZK Proof Systems

A Comparative Analysis of Modern Zero-Knowledge Proof Technologies

Group 12

Anubha
Tanmoy
Yogesh
Rosemary

Objective and Scope

Objective

- Experiment and get experience implementing zk proofs within different ecosystems
- Benchmark and compare zk proof systems.

Scope

- Evaluate systems like Halo2, Risc Zero, Jolt, etc.
- Compare metrics
- Deliver insights into their strengths and trade-offs.

ZK Proof Systems Overview

1. **Halo2:** Recursive proof system by Zcash.
2. **Risc Zero:** General-purpose ZK virtual machine, STARK inspired. Use Groth16 SNARKS for compact proofs.
3. **Jolt:** ZK-SNARK-based system.
4. **Nexus zkVM:** ZK virtual machine
5. **Circom + snarkjs:** Circuit compiler + ZK proofs.
6. **SP1:** STARK Based
7. **Powdr:** STARK based and developer friendly.

Frameworks and Backends

1. **Halo2:**
   - Frameworks: Halo2 (Rust), Arkworks (Rust)
   - Backend: Rust, pairing-based ECC
2. **Risc Zero:**
   - Frameworks: Risc Zero SDK (Rust, C++)
   - Backend: Rust/C++, general-purpose VM
3. **Jolt:**
   - Frameworks: Jolt (Rust), Winterfell (optional)
   - Backend: Rust/Go, hash-based cryptography
4. **Nexus zkVM:**
   - Frameworks: Nexus zkVM, Arkworks (optional)
   - Backend: Rust, mixed (ECC+hash)
5. **Circom + snarkjs:**
   - Circuit: Written using circom (Rust-based)
   - Proving system: Groth16, PLONK, FFLONK
6. **SP1**
7. **Powdr**

Parameters for Benchmarking

1. **Prover Time**: Time to generate a proof.
2. **Verifier Time**: Time to verify a proof.
3. **Proof Size**: Size of the proof in bytes.
4. **Memory Usage**: Memory consumption during proving and verifying.
5. **Setup Complexity**: Trusted vs. transparent setup.
6. **Supported Features**: Recursive proofs, universal circuits.
7. **Post-Quantum Resistance**: Security against quantum attacks.
8. **Scalability**: Efficiency with increased complexity
9. **Parallel execution**: Ability to parallelize proving/verifying

Cryptographic Assumptions

1. Elliptic Curve Cryptography (ECC):
   - Used in Halo2, Plonky3, Aleo.
   - Assumes the hardness of the Discrete Logarithm Problem (DLP).
2. Hash Function Assumptions:
   - Used in Miden VM, Risc Zero.
   - Assumes collision and preimage resistance.
3. Polynomial Commitment Assumptions:
   - Used in Plonk, Halo2.
4. Transparent Setup (STARKs):
   - Used in Miden VM and Risc Zero.

Operations for Benchmarking

1. Sha256
2. Fibonacci
3. Poseidon Hash

General Comparisons

Proof System	**Setup Complexity**	**Features**	**Post-Quantum Resistance**	**Scalability**	**Parallel Execution**
Halo2	Transparent generally	Recursive proofs	No (ECC based)	High	Limited
Circom (Groth16)	Trusted Setup	Efficient proofs	No (Pairing-based)	Moderate	High
Risc Zero	Transparent	General purpose	Yes	High	High
Jolt	Can support both	Efficient proofs	Yes	Very High	Very High
Nexus zkVM	Transparent	Privacy focused	Partial	Moderate	High
SP1	Transparent	rollup optimized	Yes	Very High	Very High
Powdr	Transparent	Extensible	Yes	High	High

Benchmarking Setup

Proof System	**Hardware Specification**
Halo2	i7-13700F @ 2.10 GHz, 32 GB RAM
Circom	Dell Inspiron 5570 (i5-8250U CPU @ 1.60GHz 1.80 GHz), 8 GB RAM
Risc Zero	(i5-11300H CPU @ 3.80GHz), 24 GB RAM
Jolt	Macbook M2 Pro - Core 16 - Memory 16 GB
Nexus zkVM	Macbook M1 Pro - Core 8 - Memory 8 GB
SP1	Macbook M1 Pro - Core 8 - Memory 8 GB
Powdr	AlmaLinux 8.10 - Core 16 - Memory 32 GB - Disk 1 TB

Benchmarking Results
(SHA256 - 1 KB input)

Proof System	Prover Time (s)	Cycles	Verifier Time (s)	Prover Memory (KB)	Constraints	Proof Size (B)
Halo2	14.78s	-	0.13s	1134KB	NA	4064B
Circom	46.07 s	-	1.14 s	3920848 KB	540736	805 B
Risc Zero	2.5 s	65536	NA	NA	NA	210157 B
Jolt	26.39 s	62231	0.054 s		NA	401116B
Nexus	30 + mins	NA	NA	NA	NA	NA
SP1	17.6 s	71249	0.172 s	NA	NA	2656912 B
Powdr	9.07 s	73731	NA	NA	NA	NA

Benchmarking Results
(Poseidon - 32 B input)

Proof System	Prover Time (s)	Verifier Time (s)	Prover Memory (KB)	Proof Size (B)	Constraints/ Trace Len
Halo2	8.74 s	0.086 s	25 KB	2144 B	
Circom	1.19 s	0.72 s	373560 KB	804 B	4184
Risc Zero	5.47 s	NA	NA	256742 B	524288
Jolt	434.08 s	0.24 s	NA	477746	554595
SP1	112.5	0.509 s	NA	2876912 B	39479
Powdr	21.54 s	NA	NA	NA	286652

Benchmarking Results (Fibonacci - 10000 elements)

Proof System	Prover Time (s)	Cycles	Verifier Time (s)	Prover Memory (KB)	Proof Size	Constraints
Halo2	0.196	NA	0.004	9.8	1664B	NA
Circom	1.75	NA	0.81	466280	805 B	9999
Risc Zero	6.37	65536	NA	NA	206182 B	NA
Jolt	36.79	280287	0.06	NA	452398	NA
Nexus (max input 100)	35.2	NA	2.4	NA	47.9 MB	NA
SP1	18.87	69101	0.174	NA	2656912B	NA
Powdr	8.64	2990	NA	NA	NA	NA

Visualization of Benchmarking Results

Observations and Insights

Challenges and Recommendations

Conclusion

Impact

- Make informed decisions in ZK system selection
- Paves way for optimizing zk systems for real world scenarios

Next Steps

- Standardize the benchmarks and test on diverse system environments
- Extend analysis to new ZK systems
- Explore hybrid configurations