ZKVM

A breif intruduction to zkvm' s past, present and future

Yao Wang 2025-01-01

What is a ZKVM?

ZKVM is a vm with zero-knowledge proof

ZKVM is a virtual machine that executes programs

and generates proof for their executions

About Virtual Machine

- Like CPU execute instructions, vm usually executes bytecode.
- Popular vm: JVM,Python VM WASM VM
- Purpose compile once run everywhere.
- Also named as interperter

About ZK

Zksnark: what we care and what we do not

- ZKsnark: “Zero-Knowledge Succinct Non-Interactive Argument of Knowledge.”
- What we care: succinctness
- What we do not care: zero-knowledge.

Goal for zkvm

Privacy is nice, but

- In practice, The main goal of people building zkvm is to verify the computation on the chain.
- Layer 2 mainly cares about proof speed and cost. Privacy is a non-goal for layer2 except a few.
- Zero-knowledness is often a byproduct of succinctness. (Protocol paper often has non-zk version and zk version)

Workflow of a typical zkvm

User's perspective

- Write programs in high-level language like Rust or Solidity
- Compile the program into binary with respect to the zkvm's targetted bytecode.
- Run the binary in the zkvm and collect execution trace
- Use the trace and circuit to generate the proof.
- Verify the proof elsewhere (usually on chain)

Decomposition of ZKVM

Frontend, backend and PCS

- The virtual machine and its instruction sets are usually referred as frontend
- The proof system is the backend.
  - Arithematization (Halo2, Plonk, STARK AIR HyperPlonk, MLE-based etc)
  - PCS (KZG,IPA, FRI based)
  - Recursive Prover?
  - Snarkify verifier?

Recipe for making your own ZKVM

- Pick your favorite VM: EVM, WASMVM, RISCV VM
- Pick your favorite tool to write the circuit (arithmetization of your VM)
- Pick your favorite proving system
- Put them together

ZKEVM

Where it started

- Vitalik found zkp prototol to scaling up Ethereum
- People start building layer2 by making their own “ZKEVM
  - Zksync ZKEVM
  - Starknet cairo VM (runs the Cairo language)
  - Taiko ZKEVM (Halo2 based)
  - Scroll’s ZKEVM (Halo2 based on prduction)
  - Polygon zkevm (plonky2 based)

Problems for ZKEVM

Hmmm

- EC-based proving systems are slow
- EC-base PCS are inefficient (big field for small data)
- EVM is not ZK-friendly. (Long word length Keccak hash)
- EVM instructions are very complicated

ZKVM

Better than zkevm

- People realize that you do not need to run the contract by EVM on layer2
- RISC0, the first RISCV-based zkvm
- Performant by recursive stark and small field
- Riscv-instruction set is simple and circuit is less complecated

Manymore

- Jolt
- Valida
- Nexos
- SP1
- ZKwasm- Dephiluslab

Development of ZKVM techs

- Recursive prover introduced by Plonky2 (Most “modern” zkvm use this)
- Snark vs stark. (Stark wins on Prover. Snark left for on-chain verification)
- multi-table design VS single table
- Lookup-based VS arithematic (start by Jolt paper adopted by sp1 and valida)
- Non-Cpu (OpenVM) VS cpu base (Other zkvms) (valida is kinda between)

Non-determinism!

ZKVM vs ZK circuit

- ZK circuit arithmetizes one program and proves its execution
- ZKVM arithmetizes one program (The VM!) and proves its execution.
- general purpose: program agnostic/VM overhead/Simple circuit
- Specialized: optimized circuits. Need redo circuit when program changes

Precompile for ZKVM

- When one function is in the hot path and zk unfriendly, write circuits for it and use this special circuit to prove it instead of the VM
- Most zkvm does this for keccak hash and EC group ops.

Challenges

- Faster Prover
- Verifiable compilation (source code to binary proof)
- Formal Verifiable VM circuit
- Deprecate Groth16

Reference

- A guide to Zero Knowledge Proofs https://medium.com/@Luca_Franceschini/a-guide-to-zero-knowledge-proofs-f2ff9e5959a8 Explains basics of zkp, especially IOP and PCS.
- The different types of ZK-EVMs https://vitalik.eth.limo/general/2022/08/04/zkevm.html
- Jolt https://eprint.iacr.org/2023/1217