

PDAO REPORT

2025.11.25.

From ZKP to SP1 Hypercube

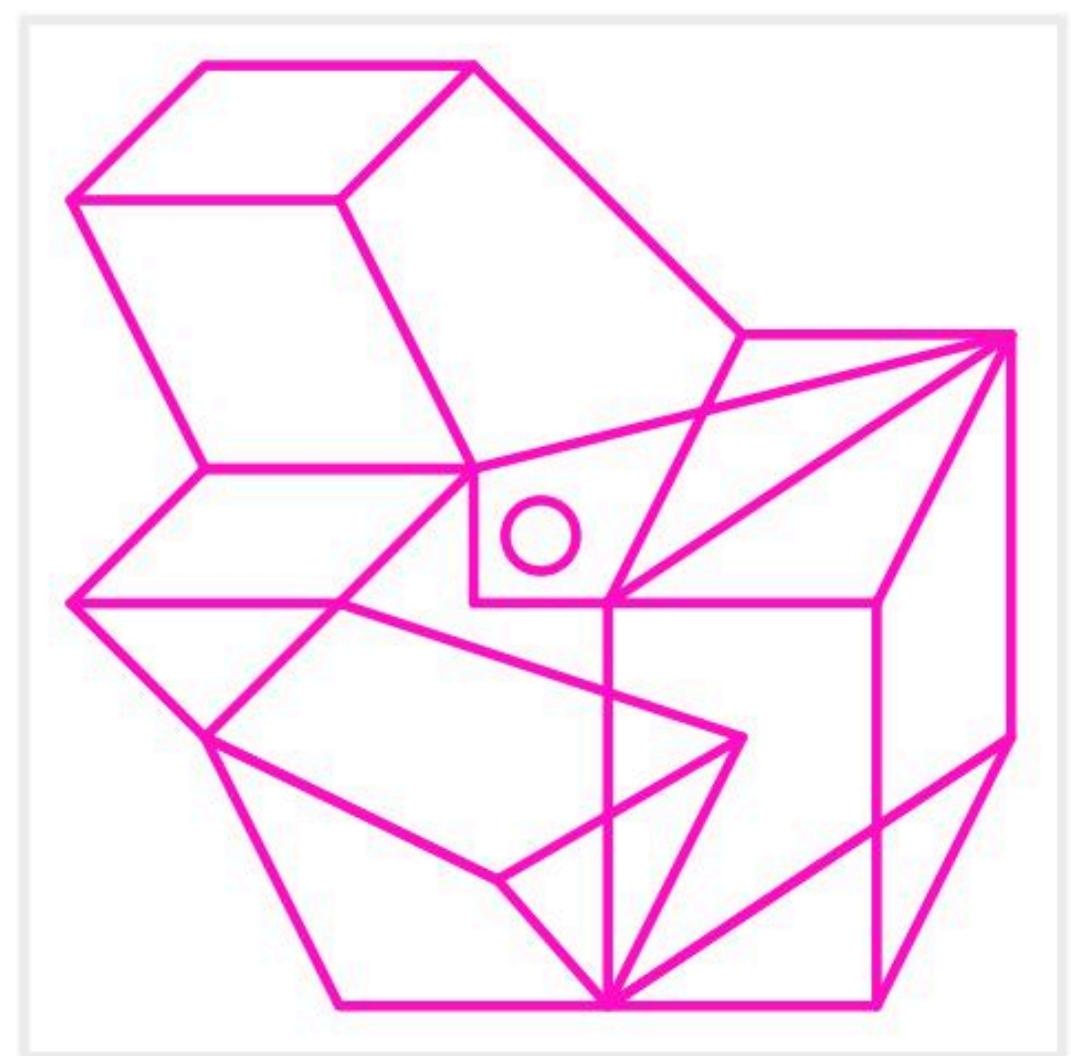
-How Succinct Achieve Real-time Proving?

ZKP

zkVM

Real-time
Proving

Jagged PCS



Written by Seungjun
X:@seungjun_x

Reviewed by fakedev9999
X:@fakedev9999





이 리포트는 2025년 10월 중 2025-2 PDAO Study에서 진행한 “ZKP부터 SP1 Hypercube까지” 아티클의 내용을 기반으로, 최신 이슈들을 함께 정리한 리포트입니다.

PDAO Study에서 진행한 내용을 아래 링크에서도 확인하실 수 있습니다.
[Youtube](#) / [Medium](#)

INDEX

Intro - 매력적인 ZK Technology

I. zkVM

- 영지식 증명(Zero-Knowledge Proof)
- ZKP가 블록체인에서 중요한 이유
- zkVM
- zkVM의 구성
- zkVM Process Flow

II. Real-Time Proving

- 현존 이더리움의 블록 확정의 한계 : 12초가 중요한 이유
- 확정 시간의 대폭 개선
- Real-Time Proving의 기준

III. Jagged PCS - Key Implementation of Succinct's SP1 Hypercube

- Succinct와 SP1 Hypercube
- ZKP 중 PCS의 역할
- 기존 PCS 처리에서의 한계점
- Jagged PCS
- Jagged PCS를 통한 검증 개선
- SOL 생태계 내 zkVM 프로덕트 사례

Outro - Exponential improvement in zkVM

Intro

1

매력적인 ZK Technology

ZKP(영지식 증명)는 크립토 산업에서 가장 매력적인 기술 중 하나로 손꼽힌다. ZKP는 블록체인 초창기부터 제기된 *블록체인 트릴레마 중 scalability에 대한 획기적인 개선의 시작점이 되었다. 특히, zk-rollup을 시작으로 Ethereum 생태계의 L2 발전에 기여하였고, 현재 이로부터 파생된 프로젝트들이 넘쳐나고 있다.

2025년 10월을 기준으로 L2Beat의 ZK Catalog에 집계된 프로젝트들의 *TVS는 약 80억 달러에 달한다. 이처럼 다양한 zk 프로덕트들이 등장한 시점에서, 이들은 “Real-Time Proving”이라는 하나의 목표를 위해 계속해서 기술을 발전해나가고 있다.

블록체인 트릴레마

블록체인 네트워크에 있어서 ‘확장성(Scalability), 보안성(Security), 탈중앙성(Decentralization)’의 세 가지 요소를 동시에 달성하기 어려운 문제를 의미한다.

TVS (Total Value Secured)

해당 체인이 확보하고 있는 총자산가치(TVL)뿐 만 아니라 L1에서 브릿징된 자산 가치까지 포함하는 개념의 총자산가치.

그림 1. ZK Catalog in L2Beat

ZK Catalog				Search
#	NAME	TVS	VERIFIERS	TECH STACK
1	SP1 Succinct	\$3.22B	1	zkVM zkVM: Plonk: Gnark Used in: Stark, Groth16, SNARK, Plonk, Linea, Lighter, Stone, STARK: Stone, Final wrap, Plonk: Plonky3, ISA: RISC-V, Field: Baby Bear, Final wrap, Plonk: Gnark, Groth16: Gnark, curve: BN254
2	Linea ConsenSys	\$1.50B	1	zkVM zkVM: Plonk: Gnark Used in: Linea
3	Lighter Lighter	\$1.14B	1	zkVM zkVM: Plonk: Gnark Used in: Lighter
4	Stone Starkware	\$1.13B	1	zkVM zkVM: Plonk: Gnark Used in: Stark, Linea, Lighter, Stone, STARK: Stone, Final wrap, Plonk: Plonky3, ISA: EVM, curve: BLS12-377, curve: BW6-761, Final wrap, Plonk: Gnark, curve: BN254

자료 : <https://l2beat.com/zk-catalog>

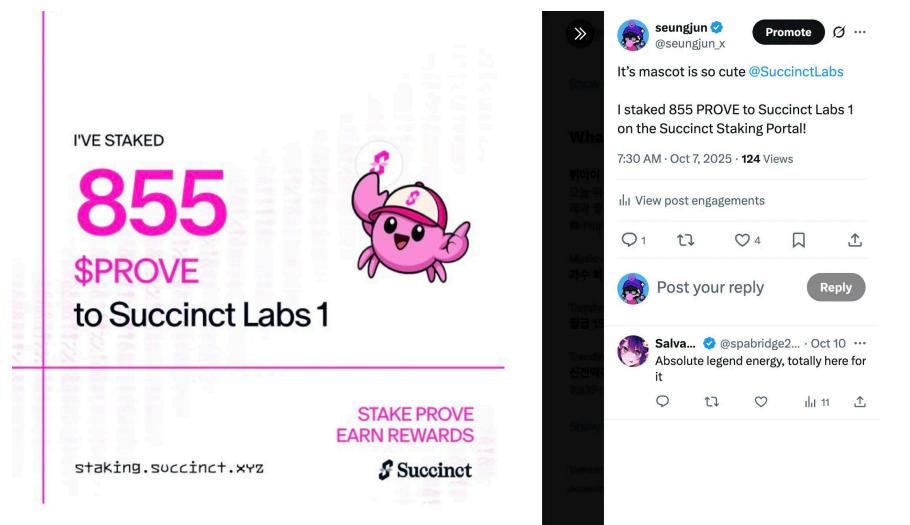
이 중에서도 가장 앞선 프로젝트가 바로 Succinct의 SP1이다. Succinct는 독창적이고 강력한 기술력을 바탕으로 SP1 Hypercube라는 zkVM을 개발하여 zk 시장을 선도하는 기업이다.

이번 리포트에서는 zk 기술에 대한 이해를 바탕으로, zk의 목표라 불리는 “Real-Time Proving”이 무엇인지, 그리고 선도 기업인 Succinct는 어떠한 방식으로 이를 구현하고자 하는지 알아본다.

Chap 1에서는 ZKP를 이해하기 위한 사례부터 시작하여, ZKP가 블록체인에서 왜 중요한지 알아본다. 그리고 이러한 ZKP를 생성하는 zkVM이 무엇인지, 어떻게 ZKP를 생성하는지 알아본다. Chap 2에서는 Ethereum에서 “Real-Time Proving”이 어떠한 의미인지 알아본다. 마지막 Chap 3에서는, “Real-Time Proving”을 구현하기 위해 Succinct의 SP1 Hypercube가 도입한 “Jagged PCS”的 아이디어를 알아본다. 이후 리포트를 마치며 SP1 Hypercube 이후로 급격한 속도로 발전하고 있는 zkVM의 발전 과정을 팔로업해본다.

필자 또한 개인적으로 Succinct에 Staking을 하고 있다. 리서처이자 한 명의 투자자로서, 난이도가 높은 컴퓨터공학적 지식 및 대수적 지식 등을 최대한 많은 투자가 이해할 수 있도록 쉽게 풀어쓰고자 했다. 다소 길다고 느낄 수 있지만, 이 글을 읽으면 투자자들이 자신이 어떤 것에 투자하고 있는지 이해하는데 도움이 될 것이다. 이 글을 통해 많은 투자가 zkVM 프로젝트들이 어떠한 여정을 가지고 발전해나가는지 이해하는 계기가 되기를 바란다.

그림 2. Succinct Staking Image



자료: Seungjun

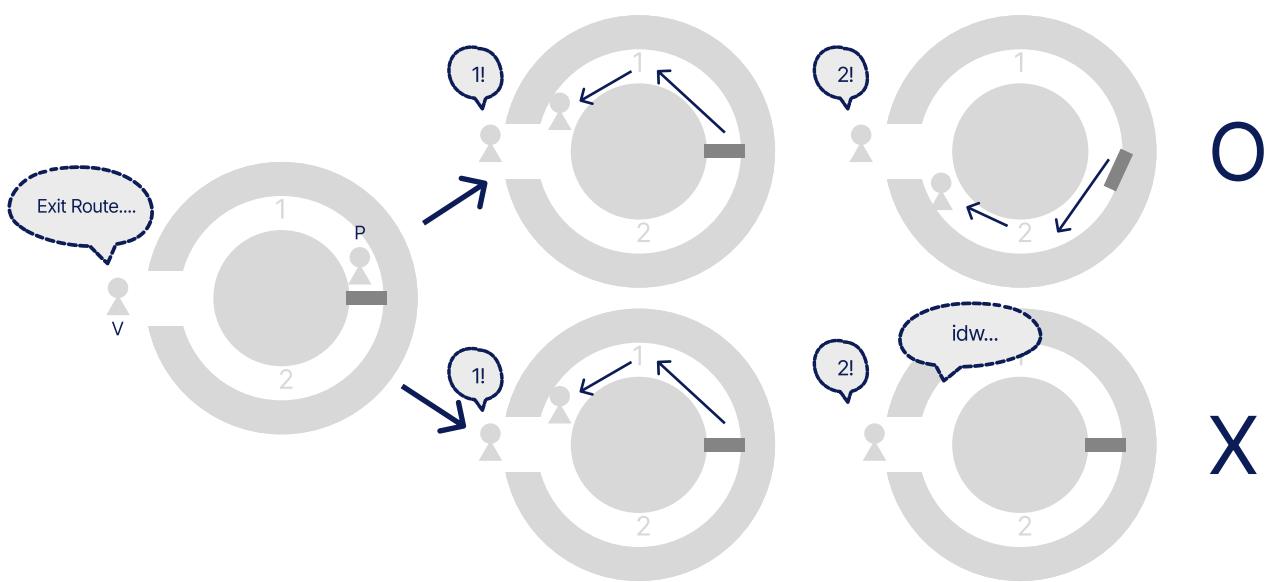
I. zkVM

2

ZKP

가장 먼저 영지식 증명이 무엇인지부터 알아보자. 영지식 증명이란, 정보(Knowledge)를 공개하지 않고서도(Zero) 그 정보를 알고 있음을 증명(Proof)하는 암호학적 기법을 통틀어 말하는 표현이다. 영지식 증명을 잘 설명하는 대표적인 예시로 “알리바바의 동굴”이 있다. 알리바바의 동굴은 이미지처럼, 하나의 입구만 존재하는 도넛 형태의 동굴이다. 그리고 이 동굴의 내부에는 도어락이 달린 문이 있다.

그림 3. 알리바바의 동굴



자료 : PDAO, Seungjun

두 사람 P(Prover, 증명자)와 V(Validator, 검증자)가 이 알리바바의 동굴 앞에 있다고 해보자. V는 P가 도어락의 비밀번호를 알고 있는지 확인하려 한다. 하지만 그 비밀번호 자체를 물어보지는 않는다. 즉, 정 보(비밀번호)는 공개하지 않으면서도 그것을 알고 있음을 증명하려는 것이다.

이를 위해 V는 동굴 입구에서 있고, P는 동굴 안으로 들어가 (V가 보지 못하게) 1번 길이나 2번 길 중 하나를 선택한다. 그 후, V는 “1번 길로 나와라” 또는 “2번 길로 나와라”라고 무작위로 명령한다. 만약 P가 비밀번호를 알고 있다면, P가 처음에 어느 쪽 길로 들어갔든지 상관없이 도어락을 열고 반대편으로 건너 갈 수 있으므로, 항상 V의 요청대로 나올 수 있다. 만약 비밀번호를 모른다면, V의 요청을 따르지 못하는 경우가 생길 것이다.

이를 한두 번만 반복한다면 P가 운이 좋게 통과할 가능성이 있다. 하지만 이를 수십 번 반복하고, 그 기간 동안 P가 모두 정확하게 V의 요청대로 나온다면, V는 P가 비밀번호를 알고 있다고 확률론적으로 확신 할 수 있다. 더 구체적으로, 데이터나 입력값을 공개하지 않고도 그 데이터에 대한 지식이나 계산 결과의 유효성을 증명하는 행위를 ZKP라고 한다.

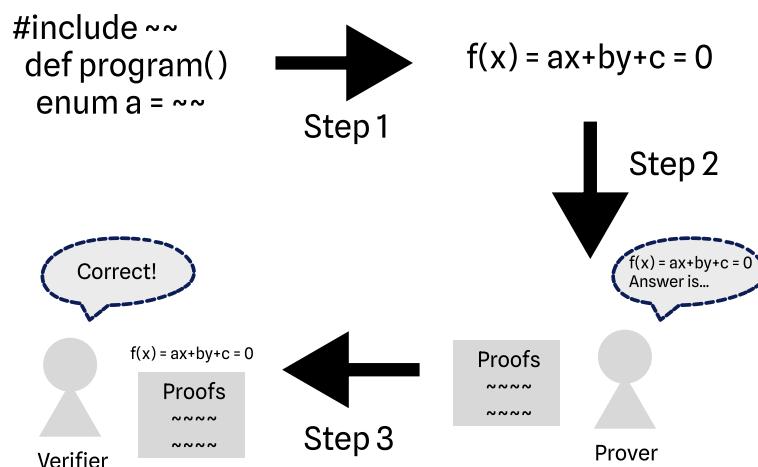
3

ZKP가 블록체인에서 중요한 이유

이 ZKP가 왜 중요하나면, 블록체인 상에 생성되는 트랜잭션들(정보)에 대해서, 이를 직접 실행(확인)하지 않고도 트랜잭션이 정상적으로 작동한다는 유효성을 확인할 수 있게 하기 때문이다. ZKP가 블록체인 위에서 구현되는 과정을 간략하게 일반화하면 다음과 같은 3가지 과정을 거친다.

1. 모든 트랜잭션을 하나의 수학 방정식으로 변환
2. 변환된 수학 방정식이 올바르다는 것을 보이는 증명 파일 생성
3. 검증자가 해당 증명 파일이 수학적으로 올바른지 확인

그림 4. Simple ZKP Process



자료 : PDAO, Seungjun

zk-STARK

(Scalable Transparent Argument of Knowledge) ZKP를 구현하는 대표적인 방식 중 하나로, 신뢰할 수 있는 초기 설정(trusted setup)이 필요 없으며 양자 컴퓨터 공격에 대한 저항성을 갖는 특징이 있다.

이때 3번 과정에서 검증자는 기존의 모든 트랜잭션을 직접 실행하여 확인하는 것이 아니라 증명 파일만 확인하면 되므로, 매우 빠르고 효율적인 처리가 가능해진다. 검증자 입장에서는 트랜잭션 데이터 저장을 위해 사용해야 하는 공간도 줄어들고, 유효성 확인에 드는 시간도 줄어든다. 이것이 ZKP가 블록체인 확장성 패러다임을 바꾼 이유이다.

현재 ZKP를 구현하는 방식에는 다양한 방법이 존재하나, zkVM이라는 분야에서는 많은 기업들이 ***zk-STARK***라는 구현 방식의 변형을 채택하고 있다. Risc Zero나 Lita와 같은 zkVM에 zk-STARK 기반 ZKP 구현 방식이 적용되어있고, 우리가 후술할 Succinct의 경우에도 기존 SP1 Turbo 모델에서 zk-STARK를 변형한 STARKish 방식을 활용하였다.

4

zkVM

zkVM은 프로그램의 실행을 영지식 증명으로 검증할 수 있는 특별한 가상 머신(VM, Virtual Machine)이다. 가상 머신(VM)이란 특정한 프로그램을 구동하기 위해 가상의 컴퓨터 환경을 만들어 주는 프로그램을 의미한다. 블록체인 생태계에서는 트랜잭션을 처리하기 위해 만들어진 특별한 프로그램이라고 말할 수 있다.

zkVM은 그중에서도 VM에서 처리하는 트랜잭션들에 대한 ZKP를 함께 생성하는 가상 머신이다. zkVM을 활용하면 컨트랙트 개발자들은 복잡한 수학과 암호학 공부를 할 필요 없이, 프로그래밍 언어로 작성한 코드를 실행하고 ZKP를 생성할 수 있다. 이는 zk 기반 환경들과의 상호작용을 훨씬 쉽게 만든다. 예를 들어 Succinct의 SP1의 경우, Rust를 활용하여 작성한 코드에 대해 ZKP를 생성한다.

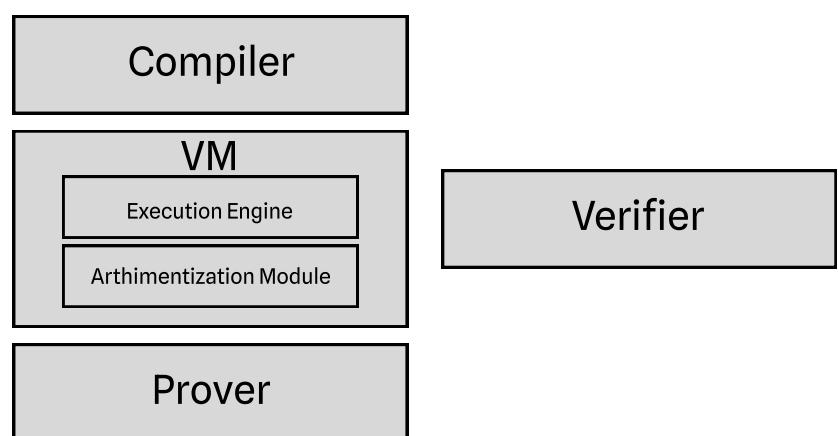
5

zkVM의 구성

zkVM을 통한 ZKP 생성 과정에는 가상 머신과 함께 컴파일러, 증명 시스템과 같은 여러 요소가 함께 상호작용한다. (일반적으로 'zkVM'은 단순한 가상 머신뿐만 아니라, 이와 상호작용하는 요소들까지 모두 포함하는 시스템을 의미한다)

zkVM이 작동하는 데 필요한 구성 요소들과 각 역할은 아래와 같다.

그림 5. zkVM의 구성요소



자료 : Seungjun

- Compiler(컴파일러): Rust 등 프로그래밍 언어로 작성된 일반 프로그램을 VM이 이해할 수 있는 머신 코드(Machine Code)로 변환한다.
- VM — Execution Engine(실행 엔진): 머신 코드로 작성된 프로그램을 실행하고, 과정 중 모든 상태 변화(입출력, 메모리 접근 등)를 추적하여 기록한다 (이를 '실행 추적'이라 한다).
- VM — Arithmetization Module(산술화 모듈): 실행 추적을 수학적 제약 조건(constraint)으로 변환하여, 산술이 가능한 형태로 만든다.
- Prover(증명자): 제약 조건들을 모두 만족한다는 것을 증명하는 영지식 증명을 생성한다.
- Verifier(검증자): 생성된 증명의 유효성을 확인(검증)한다. Verifier는 zkVM의 핵심 구성 요소는 아니지만, 생성된 증명을 최종적으로 검증하여 트랜잭션을 처리하는 주체라 할 수 있다.

ISA (Instruction Set Architecture)

명령어 집합 구조.

VM이 이해하고 실행할 수 있는 명령어
(Machine Code)의 종류와 형식을 정의한 규약

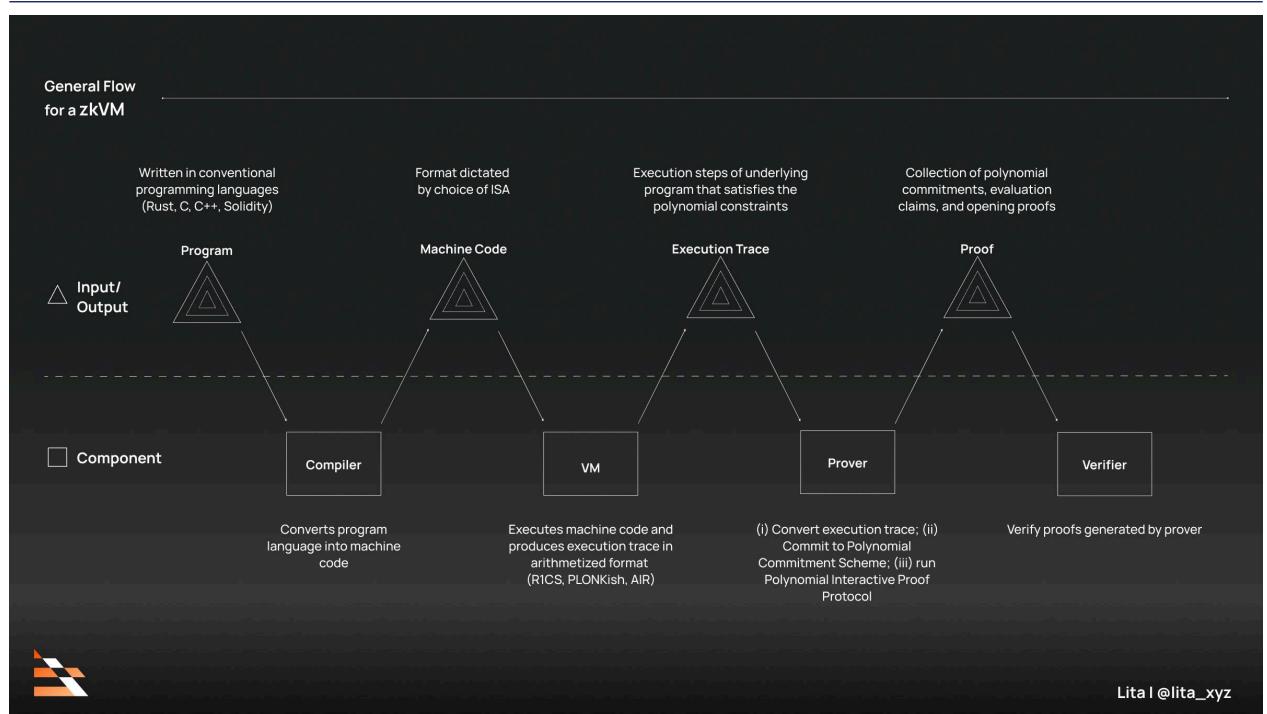
이러한 여러 구성 요소를 지니는 zkVM 시스템은, 각 구성 요소가 어떠한 방식으로 작동할지를 설정하는 방법에 따라 구현 방법과 성능, 역할 등이 달라진다. 예를 들어, 증명 시스템으로 zk-SNARK를 사용할지, zk-STARK를 사용할지, 또는 가상 머신에서 어떠한 *ISA를 채택하는지 등에 따라 달라진다.

6

zkVM Process Flow

zkVM 시스템을 구성하는 4가지 구성 요소는, 하나의 ZKP를 생성하기 위해 다음과 같은 과정을 순차적으로 거친다.

그림 6. general flow of zkVM



자료 : <https://www.lita.foundation/blog/zero-knowledge-paradigm-zkvm>

1. 컴파일러는 먼저 C, C++, Rust, Solidity와 같은 일반적인 언어로 작성된 프로그램을 머신 코드 (machine code)로 컴파일한다. 머신 코드의 형식은 선택된 ISA(Instruction Set Architecture)에 의해 결정된다.
2. VM은 머신 코드를 전달받아, Execution Engine이 이를 실행하며 실행 추적을 생성한다. 산술화 모듈은 Execution Engine이 생성한 실행 추적을 받아, 정해진 방식의 수학적 제약 조건 형태로 변환한다. (산술화)
3. 증명자는 산술화된 실행 추적을 여러 다항식 묶음의 형태로 변환한다. 즉, 실행 추적을 대수적 (algebraic)으로 표현한다. 그다음 이를 PCS에 커밋하고 (PCS에 대해서는 Chap 3에서 자세히 다룬다), 커밋된 다항식이 올바른지 보여주기 위한 증명 프로토콜을 수행하여 증명 파일을 만든다.
4. 검증자는 제약 조건 또는 커밋을 사용하여 증명 시스템의 검증 프로토콜을 따라 증명을 확인한다. 검증자는 증명의 유효성(validity)에 따라 결과를 수락하거나 거부한다.

이러한 과정을 거쳐, zkVM은 하나의 영지식 증명을 생성해낸다.

II. Real-Time Proving

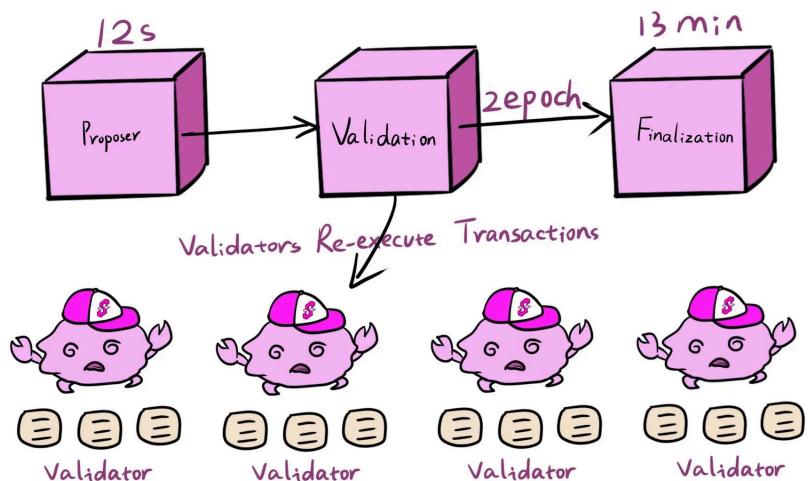
7

현존 이더리움의
블록 확정의 한계
: 12초가 중요한 이유

"Real-Time Proving"은 실시간 경험에 가깝게 증명을 생성해내는 것을 의미한다. 더 명확하게는, 이더리움 환경에서 블록 타임인 12초 이내로 증명을 생성하는 것을 말한다. 왜 블록 타임인 12초 이내로 증명을 생성하는 것이 중요할까?

그림 7. 현존 이더리움의 블록 확정

BLOCK FINALITY ON ETHEREUM



자료 : Succinct Truth Prover Writing Topics

확정(Finality)

블록체인에서 트랜잭션이 기록된 블록이 변경되거나 되돌릴 수 없는 상태가 되었음을 의미

그 이유는 현재 이더리움의 *확정(Finality) 방식과 관련이 있다. 현재 이더리움은 트랜잭션 유효성 검사를 위해 대부분의 노드들이 트랜잭션을 직접 실행해야 하며, *무신뢰성(trustless) 원칙에 따라 블록 생성과 동시에 그 블록을 확정할 수 없다.

무신뢰성(trustless)

중앙 관리 기관이나 중개자(신뢰할 수 있는 제3자) 없이도 네트워크 참여자들이 거래의 유효성을 신뢰하고 검증할 수 있음을 의미

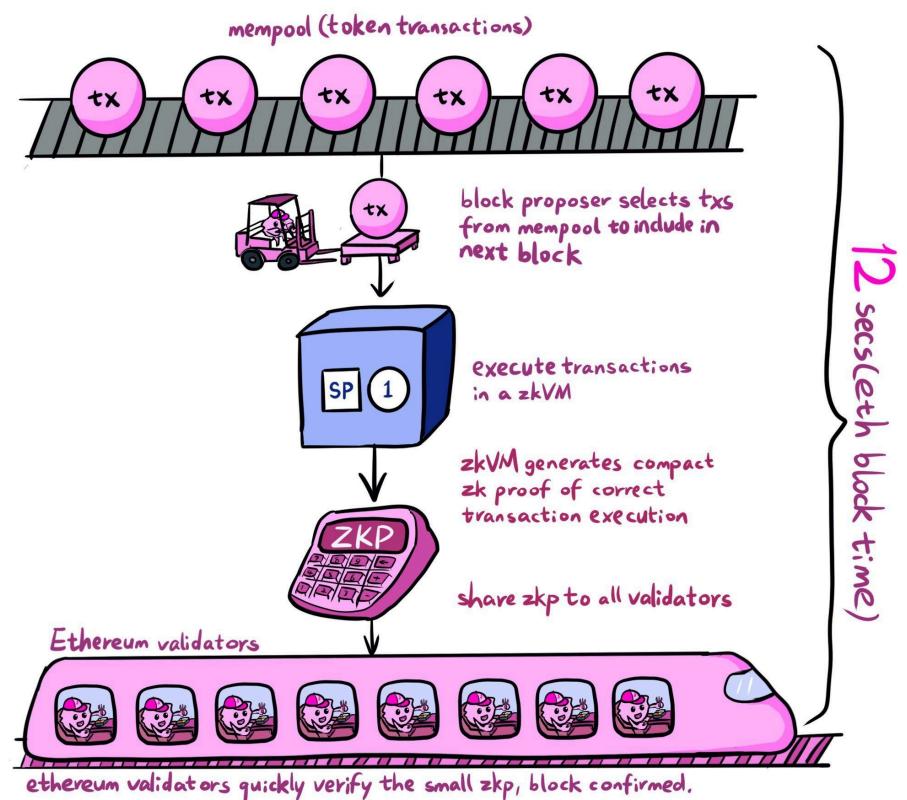
현재 이더리움은 32개의 블록(Slot)을 묶어 하나의 Epoch로 정의하며, 2개의 Epoch가 지나야 해당 블록이 확정(Finalized)된다. 이는 하나의 트랜잭션이 최종적으로 확정되는 데 약 13분(32 slots * 12초/ slot * 2 epochs ≈ 12.8분) 정도가 소요된다는 뜻이다.

확정 시간의 대폭 개선

하지만, 만약 ZKP 생성을 12초 안에 해낼 수 있다면 어떨까? 아래의 일련의 과정을 한 블록이 생성되는 시간 내에 실현할 수 있다고 가정해보자.

그림 8. Real-Time Proving의 도식화

REAL-TIME ETHEREUM PROVING



자료 : Succinct Truth Prover Writing Topics

1. *Mempool에서 블록 생성자들이 트랜잭션들을 선택하여 다음 블록에 추가한다.

Mempool (멤풀)

(트랜잭션 풀) 블록체인 노드가 수신했지만 아직 블록에 포함되지 않은 트랜잭션들의 대기 공간.

2. zkVM을 통해 트랜잭션을 실행하고, ZKP를 생성한다.
3. ZKP를 검증자(Validator)들에게 전파한다.
4. 검증자들이 ZKP를 빠르게 검증한다.

만약 이 4가지 과정이 한 블록 타임(Block Time) 안에서 일어난다면, 우리는 지금처럼 2개의 Epoch를 기다릴 필요 없이 매 블록이 생성될 때마다 해당 블록을 확정할 수 있다. 그렇다면 지금보다 64배(2 Epochs * 32 Slots/Epoch)나 빠르게 블록을 확정지을 수 있게 된다. 이것이 바로 ZKP 생성을 12초 이내에 해내는 것이 중요한 이유이며, "Real-Time Proving"의 기준이 되는 이유이다.

물론 현재의 이더리움의 확정 방식은 합의 알고리즘 상의 규약이기에, 이것이 가능해지려면 하드 포크 수준의 대규모 업그레이드가 필요하다. 현재 Ethereum Foundation은 점진적인 준비 과정으로써 현재는 일부 검증자만 zk 클라이언트를 선택적으로 실행하되, 이 비율이 높아져 다수의 검증자가 zk 클라이언트를 채택할 시 재실행 검증 방식에서 ZKP를 통한 검증 방식으로의 컨센서스 전환을 상정하고 있다.

9

Real-Time Proving의 기준

지난 2025년 7월, Ethereum Foundation은 "Real-Time Proving"에 대한 명확한 기준을 제시했다. 우리는 앞에서 시간에 대해서만 이야기를 하였지만, 이 실시간 증명의 최종 목표는 누구나 집에서 이더리움 네트워크의 검증에 참여할 수 있는 "Home Proving"이기에, 실행 컴퓨팅 자원 등에 대해서도 기준을 함께 제시하였다.

컴퓨팅 자원 기준은 실질적으로 가정에서 구동할 수 있는 수준을 목표로 한다. 현재 zkVM을 가동하기 위해서는 수십 개의 고성능 그래픽 카드 등을 사용하여 많은 컴퓨팅 자원을 소모해야 한다.

그림 9. Real-Time Proving의 기준

This brings us to our working definition of realtime proving:

- **Latency: <= 10s for P99 of mainnet blocks**
- **On-prem CAPEX: <= 100k USD**
- **On-prem power: <= 10kW**
- **Code: Fully open source**
- **Security: >= 128 bits**
- **Proof size: <= 300KiB with no trusted setups**

자료 : <https://blog.ethereum.org/2025/07/10/realtme-proving>

여기서 증명 생성의 latency가 블록 생성 시간인 12초가 아니라 10초 이내로 설정된 이유는, 일반적으로 네트워크상 데이터 전파에 소요되는 시간이 최대 약 1.5초 정도이기 때문이라고 한다. 여기서 추가로 유의할 점은, 최악의 경우(Worst Case)에도 실시간(real-time)에 준하도록 노력해야 한다는 것이다. Vitalik은 지난 5월 본인의 X를 통하여, 안전한 L1 사용을 위해서는 최악의 경우에도 “Real-Time Proving”이 필요하다고 언급하였다.

그림 10. Real-time Proving에 관한 Vitalik의 트윗



vitalik.eth 
 @VitalikButerin



1. This is average case, not worst case. We need real-time worst case for safe L1 use
2. Not formally verified
3. ~100 kW to prove. Proving is a 1-of-n trust model, but even still, perhaps we want proving doable at home (~10 kW)
4. We wanna 10-100x the L1 gaslimit

So, truly amazing work by [@pumatheuma](#) and team, but definitely still a few steps to the final destination.

1:44 PM · May 21, 2025 · 224.3K Views

자료 : X (@VitalikButerin)

III. Jagged PCS

Key implementation of Succinct's SP1 Hypercube

Succinct와 SP1 Hypercube

Succinct는 “Prove the world’s software”라는 캐치프라이즈와 함께 글로벌 zkVM 개발의 선두를 달리고 있는 프로젝트 중 하나이다. 다양한 체인에 적용 가능한 범용 zkVM 라이브러리인 SP1을 제공하고, 이를 계속해서 고성능으로 발전시키는 연구를 수행하고 있다.

그림 11. SP1 Hypercube와 Multilinear Polynomials



자료 : <https://blog.succinct.xyz/sp1-hypercube/>

이들은 지난 5월 20일, 새로운 zkVM 머신인 SP1 Hypercube를 발표하며, “Jagged PCS”라는 새로운 형태의 스키마를 제시하였고, 이 새로운 스키마를 활용하는 “Multilinear Polynomials(다중 선형 다항식)” 연산 패러다임을 소개했다.

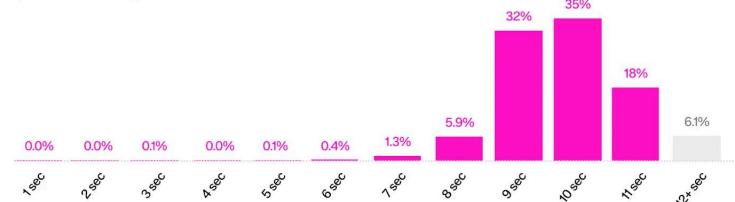
그들은 “Real-Time Proving”的 벤치마크(Benchmark)를 90% 이상의 블록이 12초 이내에 생성되는 것으로 잡았고, SP1 Hypercube는 이를 확실히 총족하였다. (93% 이상의 블록이 12초 안에 생성, 평균 블록 생성 시간 10.3초)

비록 이는 이후 Ethereum Foundation이 제시한 벤치마크에는 미치지 못하지만, 당시 기준으로는 “Real-Time Proving”에 가장 가까운 구현이라고 할 수 있다.

그림 12. SP1 Hypercube의 증명 생성 속도

Proving Latency on Ethereum Blocks

April 19th 2025 - May 3rd 2025 (200 NVIDIA RTX 4090 GPUs)



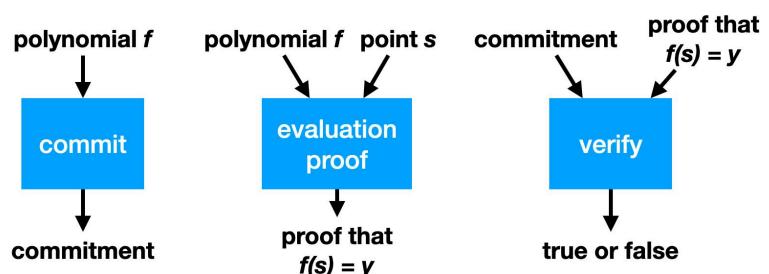
자료 : <https://blog.succinct.xyz/sp1-hypercube/>

ZKP 중 PCS의 역할

Succinct가 어떻게 zkVM의 연산 속도를 개선할 수 있었는지 알아보기 위해, Jagged PCS라는 스키마가 어떤 것인지 알아볼 것이다. 우선 PCS가 무엇인지 알아보기 위해, Chap 1에서 알아보았던 zkVM의 증명 생성 프로세스를 복습해보자.

1. 트랜잭션을 컴파일러가 머신 코드로 변경하고, VM은 이를 실행하며 실행 추적을 기록한다.
2. 실행 추적을 산술화 모듈이 산술적 제약 조건으로 변환한다.
3. 변환된 산술적 제약 조건은 Prover에게 넘어가고, 이것을 대수적인 다항식으로 변환한다. 이때 생긴 다항식들을 “커밋”한다.

그림 13. Commit, Proof, Verify in ZKP



자료 : https://o1-labs.github.io/proof-systems/plonk/polynomial_commitments.html

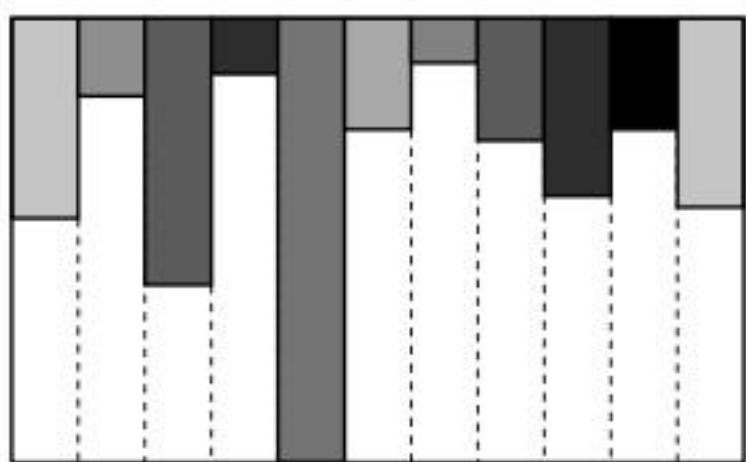
이 다항식들을 커밋하는 기법을 PCS(Polynomial Commitment Scheme)라고 한다. 여기서 말하는 ‘커밋(Commit)’이란 다항식의 암호학적 요약본(fingerprint)을 생성하는 것을 의미하며, 이는 다항식을 미리 처리하여 간결하게 만든 버전이라고 할 수 있다. 이렇게 생성된 ‘약정(commitment)’은 이후 단계에서 검증자에게 전달되어 Proof 파일의 유효성 검증에 활용된다.

기존 PCS 처리에서의 한계점

기존 zkVM에서 Prover는 실행 추적을 대수적인 다항식으로 변환한 후, 이 여러 개의 다항식을 계산 정확성을 위해 여러 개의 테이블로 나타낸다. 이 테이블은 서로 다른 CPU 명령어에 해당한다.

이렇게 생성된 여러 테이블에서, 테이블 각각의 개별 열이 분리된 다항식으로 간주되어 커밋을 수행하고 Proof 파일을 검증자에게 제공한다.

그림 14. The sparse polynomials committed to in the jagged PCS



자료 : Jagged polynomial commitments

오버헤드(overhead)

어떤 작업을 처리하는 데 있어 직접적인 연산 외에 부가적으로 소요되는 시간, 메모리, 연산 자원.

이 방식은 여러 다항식의 크기(회색 사각형)가 제각각이므로, 임의의 빈 공간을 의미 없는 0으로 채워 넣은(흰 공간) 희소한(Sparse) 형태의 PCS를 생성한다. 이렇게 개별 열을 분리된 다항식으로 커밋하는 방식은 연산 과정에서 심각한 *오버헤드를 초래한다.

첫째, 증명자가 열들을 분리하여 인코딩하는 한, 검증자는 시스템에 포함된 열의 수에 비례하여 선형적으로 증가하는 상당한 양의 작업을 수행해야 한다.

둘째, 기존 설계는 테이블의 높이와 너비가 모두 2의 거듭제곱(power of two)이어야 한다는 제약이 따른다. 이는 이미지와 같이 테이블에 불필요한 빈 공간(zero-padding)을 생성하여 연산의 복잡성을 야기한다.

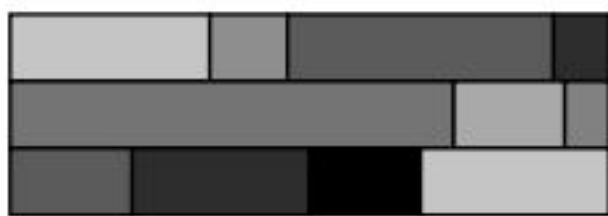
셋째, 가장 중요한 문제로, PCS가 생성한 여러 테이블의 높이가 일정하지 않다는 점이다. Prover는 증명 파일을 생성하는 과정에서 'recursion circuit'이라는 회로를 사용하는데, 이 회로는 테이블의 높이에 따라 그 구조가 달라진다. 즉, 높이가 다른 테이블이 여러 개 존재하면, recursion circuit의 수 또한 그만큼 필요하게 되며 잠재적으로 요구되는 회로의 수가 폭발적으로 증가하게 된다. 이를 "조합 폭발" 문제라고 한다.

이처럼 현재의 PCS 구조는 연산 효율에 있어 최적화할 여지가 많은 형태로 존재한다.

Jagged PCS

Succinct는 이러한 비효율적인 PCS의 구조를 최적화하기 위해 "Jagged PCS"를 제안한다. 이는 기존의 'Jagged'(들쭉날쭉한) 테이블, 즉 희소한(sparse) PCS 테이블 구성을 이미지와 같이 'dense'(밀집된) 형태처럼 효율적으로 매핑하여, 논리적인 단일 구조로 커밋할 수 있도록 하는 방식이다.

그림 15. The dense form of the data



자료 : Jagged polynomial commitments

즉, Prover는 실제 실행 추적 정보에 해당하는 다항식 정보(각기 다른 길이를 가진 열들)만을 빈틈없이 모아, 이를 하나의 밀집된 다항식처럼 '포장'하여 단 한 번의 커밋으로 처리할 수 있다.

이를 실현하는 과정을 간단히 설명하자면 아래와 같다.

- 1. 다항식 압축:** Prover는 2차원 배열 형태인 Sparse PCS에서 실제 데이터만을 순서대로 정렬하여, 하나의 단일 다항식(1차원 배열)으로 압축한다.
- 2. 단일 커밋 및 누적 높이 생성:** Prover는 이 단일 다항식에 대해서만 커밋을 수행한다. 이와 함께 원본의 각 다항식(열)이 압축된 배열의 어디서부터 시작하는지 알려주는 누적 높이(cumulative heights, 각 열의 높이를 순서대로 더한 값의 시퀀스)를 계산하여 검증자에게 공개한다.
- 3. 위치 변환:** 검증자가 기존 PCS 형태의 정보(2차원 배열상의 위치)를 요청하면, Prover는 이 누적 높이를 참조하여 압축된 배열에서의 위치를 계산한다.
- 4. 단일 증명 및 검증:** Prover는 이 압축된 배열의 해당 위치 값에 대해 증명 파일을 생성하고, 검증자는 이 단일 증명만을 검증한다.

특히, 3번과 4번의 과정을 통하여 다른 특수한 형식의 아키텍처를 사용하지 않고도, 기존 검증자의 검증 과정을 그대로 사용할 수 있다는 점이 중요하다.

Jagged PCS를 통한 검증 속도 개선

시간 복잡도(Time Complexity)

'선형(Linear, $O(n)$)'은 데이터의 수에 비례해 검증 시간이 증가함을, '상수(Constant, $O(1)$)'는 데 이터 수와 관계없이 검증 시간이 일정함을 의미.

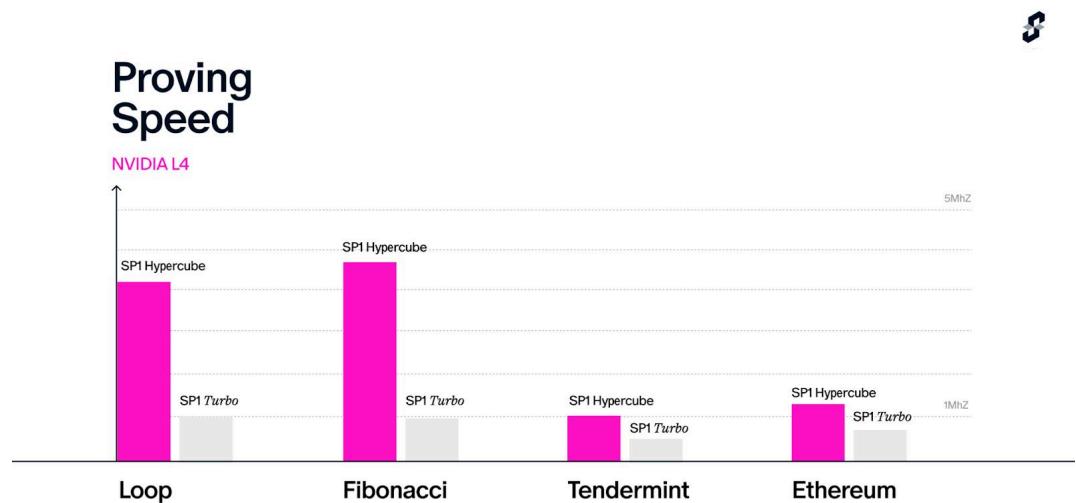
이러한 Jagged PCS는 기존의 PCS 구조를 통한 연산에 비해 다음과 같은 두 가지 이점을 제공한다.

첫째는 검증자의 오버헤드 해결이다. 기존 방식에서는 검증자가 각각의 다향식에 대한 커밋을 개별적으로 확인해야 했기에, 이 작업량이 열의 개수에 비례했다. 하지만 Jagged PCS는 이를 하나의 밀집 다향식에 대한 커밋으로 변경하였다. 이는 검증의 *시간 복잡도를 선형(linear)에서 상수(constant) 수준으로 획기적으로 낮추는 계기가 된다. 또한, 이는 검증 비용 절감으로도 이어지는데, Succinct는 이를 "pay only for what you use" architecture라고 설명하였다.

둘째는 조합 폭발 문제의 해결이다. 기존 방식에서는 PCS 테이블의 높이가 달라지면, 이를 검증하기 위해 서로 다른 검증 회로를 사용해야 했으며, 이로 인해 필요한 검증 회로의 수가 늘어나는 "조합 폭발" 문제가 있었다. 하지만 Jagged PCS에서는 이미 크기가 알려진 하나의 밀집 다향식과 누적 높이 (cumulative heights)만이 필요하므로, 검증자는 동일한 검증 회로를 사용하여 "조합 폭발" 문제를 해결할 수 있게 되었다.

이러한 장점을 지닌 Jagged PCS를 적용한 SP1 Hypercube는 기존의 자사 모델인 SP1 Turbo에 비해 2배 ~ 최대 5배 정도의 높은 연산 속도를 보여준다.

그림 16. SP1 Hypercube의 성능 비교



자료 : <https://blog.succinct.xyz/sp1-hypercube/>

Outro

15

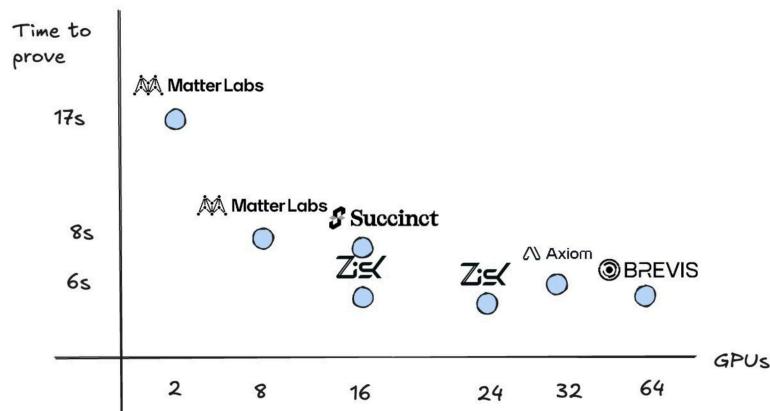
Exponential improvement in zkVM

zkVM이 어떤 것인지부터, 이를 활용하는 이더리움의 “Real-Time Proving” 로드맵, 그리고 이에 최초로 다가간 SP1 Hypercube에 대해서 알아보았다. SP1 Hypercube가 나온 지 반년 정도밖에 지나지 않았지만, 그사이 zkVM 시장은 치열한 경쟁 속에서 지수적인 발전을 이루고 있다.

SP1 Hypercube가 160개의 RTX 4090으로 12초 이내 L1 블록의 94%를 증명한 지 5개월이 지난 10월 14일, Brevis 사의 zkVM 모델 Pico Prism은 64개의 RTX 5090으로 12초 이내 L1 블록의 99.9%를 증명하는데 성공했다.

11월 18일에 Succinct는 SP1 Hypercube를 통해 16개의 RTX 5090만으로 12초 이내 L1 블록의 99.7%를 증명하는데 성공했다고 밝혔다. 그리고 11월 21일, zkSync 사의 zkVM 연구팀 Airbender는 DevConnect 기간 동안 단 두개의 RTX5090만을 활용하여 평균 17초 정도의 latency로 이더리움 블록 증명에 성공했음을 밝혔다.

그림 17. 현 zkVM 프로덕트들의 GPU 사용량과 latency 비교 그래프



자료 : <https://x.com/BowTiedGolem/status/1991889549497417853>

이는 단순히 GPU의 성능 발전에서만 나오는 성과가 아니라, 각 팀들의 더 빠른 증명을 위한 엄밀한 연구들로부터 나타나는 성과이다. 지난 11월 13일 Succinct 사는 Jagged polynomial commitments에 이어 TensorSwitch라는 새로운 최적화된 PCS 디자인에 대한 논문을 공개하였다.

그림 18. TensorSwitch

Paper 2025/2065

TensorSwitch: Nearly Optimal Polynomial Commitments from Tensor Codes

Benedikt Bünz , New York University, Espresso Systems

Giacomo Fenzl , École Polytechnique Fédérale de Lausanne

Ron D. Rothblum , Succinct

William Wang , New York University

자료 : <https://eprint.iacr.org/2025/2065>

현존하는 zkVM 프로젝트 현황은 <https://ethproofs.org/>에서 확인할 수 있다. 이 대시보드를 통해 Airbender, Jolt, OpenVM, Pico Prism, SP1 Hypercube, Ziren 등 여러 프로젝트가 서로 경쟁하고 있음을 알 수 있다.

그림 19. ETHProofs



자료 : <https://ethproofs.org/>

또한, 오는 12월에는 이더리움의 Fusaka 업그레이드가 계획되어 있다. 이는 이더리움 가스 관련 조정을 통해 증명 비용에 긍정적인 영향을 미칠 업그레이드로 기대된다. 이를 통해 Ethereum Foundation이 제시한 "Real-Time Proving" 목표에 거의 다다르는 획기적인 발전이 이루어질 것으로 보인다.

집에서 가정용 PC로 누구나 이더리움 증명에 참여할 수 있는 날("Home Proving")이 머지 않았다.



PDAO

PDAO는 POSTECH(Pohang University of Science and Technology)을 거점으로 한 크립토-블록체인 커뮤니티입니다.

PDAO는 2022년 초 POSTECH의 크립토 커뮤니티이자 오픈소스 개발 그룹으로 시작되어, “DAO를 개발하는 DAO”라는 정체성을 바탕으로, 탈중앙화 DAO 프로토콜인 Simperby를 개발하며 이를 통해 스스로를 호스팅해왔습니다. 이외에도 멤버들의 각종 해커톤 참여 및 2023년 과기부 산하 오픈소스 컨트리뷰션 프로젝트 등 개발조직으로써 블록체인 산업에 기여해왔습니다.

2025년 현재 PDAO는 누구나 자유롭게 참여할 수 있는 커뮤니티로써, 블록체인 생태계에 관심있는 사람들이 자유롭게 이 산업에 기여할 수 있는 여러 활동들을 기획하고 참여할 수 있는 단체로 운영되고 있습니다. 이에 2025년 상반기 Superteam KR Guild Lead, Monad Blitz Seoul Supporters 활동 등 각종 재단 및 기업과 각종 프로젝트를 기획, 운영, 참여하고 있습니다.

Official Website : <https://dao.postech.ac.kr>

X : @postech_dao

Reviewed by

fakedev9999

X : @fakedev9999

Simperby Core Team Head, PDAO (2022 -)
Software Engineer, Succinct (2025 -)

Written by

Seungjun

X : @seungjun_x

Organizer (Head), PDAO (2025 -)
Researcher, Bithumb (2022-2023)

Disclaimer

- 본 리포트는 투자 결과에 대한 법적 책임 소재의 증빙자료로 사용될 수 없습니다.
- 투자에 관한 모든 결정과 책임은 투자자 본인에게 있습니다.
- 본 리포트는 신뢰할 만한 자료 및 정보를 토대로 작성되었으나 그 정확성에 대해 보장하지 않습니다.
- 본 자료의 저작권은 PDAO에 있으며, 출처 표기 하에 자료의 일부의 배포를 허용합니다.
- 단, 어떠한 경우에도 조직의 동의 없이 자료의 전체를 복제 및 재배포 할 수 없습니다.