Brevis ProverNet 백서

Brevis

v2.0

초록.

Brevis ProverNet 은 zkVM 프로그램부터 ZK 코프로세서 쿼리 및 집계 증명에 이르기까지 다양한 영지식(ZK) 워크로드를 이기종의 증명자 공급과 매칭하는 탈중앙화 마켓플레이스입니다. 두 가지 기둥이 Brevis 를 차별화합니다. 첫째, Brevis 는 이기종 증명과 합성, 다단계 증명 파이프라인을 처음부터 지원하도록 설계되어, 애플리케이션이 단일 스택에 얽매이지 않고 증명 시스템과 워크플로우를 혼합하고 매칭할 수 있습니다.

둘째, Brevis 는 진실성, 예산 균형, 점근적 효율성을 갖춘 양면 이중 경매인 TODA 를 도입합니다. TODA 는 지속적인 경매 라운드를 통해 증명 작업을 증명자와 매칭하며, 요청자와 증명자 모두의 무허가 참여를 보장하면서 증명 유형 전반의 수요와 공급을 청산합니다.

이기종 워크로드에 대한 강조는 경험적으로 동기가 부여되었습니다. 네트워크 설계는 다양한 증명 요구사항을 가진 애플리케이션 전반에서 운영되는 Brevis 인프라의 운영 경험을 바탕으로 하며, 이는 여러 증명 유형과 다단계 파이프라인에 대한 네이티브 지원의 필요성을 보여줍니다.

성능과 개방성의 균형을 맞추기 위해, Brevis 는 추가로 전용 롤업에 경매 로직을 배포합니다. 자체 롤업에서 마켓 컨트랙트를 실행하면 경매 처리량을 L1/L2 혼잡으로부터 격리하는 동시에 결제, 매칭, 슬래싱을 투명하고 무허가적으로 유지합니다. 이 전용 롤업은 오직 증명에 대한 입찰, 할당, 지불을 조정하기 위해서만 설계되었으며, 증명 자체는 임의의 대상 체인을 목표로 할 수 있습니다.

1. 소개

1.1. 블록체인 컴퓨팅의 현재 한계

블록체인은 "세계 컴퓨터"로 구상되었지만, 그 베이스 레이어는 여전히 제약이 많습니다. 각 트랜잭션은 합의, 보안, 결정성을 유지하기 위해 모든 검증 노드에 의해 재실행되어야 합니다. 강력한 무결성을 부여하는 바로 그 속성(완전한 복제)이 또한 강력한 성능 상한을 부과합니다: 체인은 정상 상태에서 동일한 워크로드를 실행하는 단일 머신을 능가할 수 없으며, 그 비용은 규모의 경제가 아닌 복제본 수에 비례하여 확장됩니다. 실제로, 이 복제-실행 모델은 무겁고, 상태 기반의 분석 및 개인화된 로직(예: 사용자별 할인 등급)을 온체인에서 엄청나게 비싸거나 운영상 취약하게 만듭니다.

구체적인 예는 현대 탈중앙화 거래소(DEXes)의 'VIP 트레이더' 경험입니다. 동적 수수료나 로열티 프로그램을 위한 훅은 "이 주소의 지난 30 일간 이 풀에서의 거래량은 얼마였는가?" 또는 "이 트레이더가 할인을 받을 자격이 있는 거버넌스 토큰 단위를 충분히 보유하고 있는가?"와 같은 질문을 해야 합니다. 이러한 기록을 온체인에 저장하고 재계산하는

것은 불가능합니다; 컨트랙트 내에서 체인을 스캔하는 것은 비용이 엄청나게 많이 듭니다; 그리고 암호학적 책무성 없는 오프체인 서버는 신뢰를 다시 도입하게 됩니다. 따라서 프로덕션 시스템은 다른 기본 요소를 요구합니다.

근본적인 원인: 복제된 실행.
모든 정직한 노드는 동일한 입력에 대해 동일한 상태 전환 함수를 재생하고 동일한 사후 상태에 도달해야 합니다. 이는 강력한 합의를 보장하지만 중복 작업에 컴퓨팅 자원을 낭비합니다. 공격적인 L1/L2 스케일링조차도 근본적인 비대칭을 변경하지 못합니다: 검증자가 재실행해야 하는 한, 무거운 워크로드는 전통적인 컴퓨팅 환경에 비해 비싸고 느리게 남아있을 것입니다.

1.2. 영지식을 통한 검증 가능한 컴퓨팅

실행과 검증을 분리하는 실용적인 방법은 검증 가능한 컴퓨팅을 사용하는 것입니다: 하나의 머신(또는 소규모 클러스터)이 오프체인에서 무거운 계산을 수행한 다음, 그 결과가 정확하다는 간결한 암호학적 증명을 제공합니다. 블록체인은 워크로드를 재실행하는 대신 증명을 검증합니다. 현대의 영지식(ZK) 증명 시스템은 검증 비용이 적고 예측 가능하며, 증명은 특화된 하드웨어에서 확장될 수 있으므로 이 접근 방식을 경제적으로 매력적이게 만듭니다.

ZK 시스템은 산술화를 통해 프로그램을 제약 조건으로 변환하고 다항식 커밋 스킴 및 인터랙티브 오라클 증명(IOP) 기술을 사용하여 작동합니다. 검증자는 종종 단 몇 번의 쿼리와 작은 크기의 상태만으로 정확성을 확인합니다. 검증 시간은 원본 계산 크기에 준선형적 (그리고 많은 시스템에서는 본질적으로 상수)입니다. 이를 통해 체인은 합의를 단순하게 유지하면서 임의로 복잡한 분석 및 정책 로직을 오프체인 증명자에게 아웃소싱할 수 있습니다.

DEX 훅 및 VIP 프로그램의 맥락에서, 기록 스캔, 등급 계산, 자격 확인은 오프체인에서 실행되며 증명은 훅 컨트랙트로 다시 공급됩니다. 그러면 컨트랙트는 증명을 검증하고 그에 따라 상태를 업데이트하며, 불투명한 서버를 신뢰하거나 온체인에서 기록을 재계산하기 위해 비용을 지불할 필요가 없습니다. Uniswap v4 및 기타 생태계는 바로 이 패턴의 프로토타입과 출시를 선보였습니다.

1.3. Brevis 제품: Pico zkVM 과 ZK 데이터 코프로세서

Brevis 는 두 가지 상호 보완적인 제품을 통해 검증 가능한 컴퓨팅을 운영합니다:

- **Pico zkVM** — 범용 프로그램을 효율적으로 증명하고 재귀적 집계와 구성되도록 설계된 모듈식, 고성능 영지식 가상 머신입니다. Pico 는 처리량이 높은 암호학적 "코프로세서" 및 가속 경로를 갖춘 최소한의 코어를 강조하여, 개발자가 하드웨어 이점을 활용하면서도 안정적인 VM 을 대상으로 할 수 있게 합니다.
- **ZK 데이터 코프로세서** — 지원되는 네트워크 전반의 과거 온체인 데이터를 읽고, 오프체인 계산(예: 시간 가중 잔액, 기간별 거래량, 포지션 분석)을 수행하며, 유효성 증명과 함께 결과를 반환하는 프로덕션 등급의 파이프라인입니다. 컨트랙트는 이러한

증명을 직접 소비하여, 외부 서버를 신뢰하지 않고도 VIP 트레이딩 프로그램, 로열티 보상, 수수료 할인, 능동적 유동성 정책과 같은 기능을 활성화합니다.

이러한 구성 요소는 DeFi, L2 생태계, 지갑 전반의 실제 파트너십에 배포되어 있습니다: VIP 수수료 등급을 위한 Uniswap v4 훅 디자인; PancakeSwap 및 QuickSwap 의 동적 수수료 프로그램; Euler 의 Incentra 기반 무신뢰 보상; 그리고 수백만 개의 증명과 수만 개의 고유 주소가 포함된 Linea 전반의 성장 프로그램.

1.4. 공개 마켓플레이스의 동기 및 이기종성 문제

프로덕션 배포는 중요한 현실을 드러냈습니다: 워크로드는 다양하다는 것입니다. 어떤 작업은 지연 시간이 짧고 상호작용별로 발생합니다(예: VIP 가격 책정을 위한 "스왑 전" 확인). 다른 작업은 주기적이고 대규모 배치입니다(예: 많은 주소에 걸친 주기별 인센티브 분배). 또 다른 작업은 무거운 VM 실행, 재귀적 집계, 또는 시스템 간 호환성을 위한 래핑/필드 변환입니다. 하드웨어 프로필 또한 다양합니다: 단일 GPU 장비, 다중 GPU 서버, 빠른 상호 연결을 갖춘 이기종 클러스터, 또는 혼합된 CPU/GPU/FPGA 설정. 단일 중앙화된 증명자는 이 스펙트럼을 효율적으로 감당할 수 없으며 신뢰할 수 있는 중립성을 제공할 수도 없습니다.

실제 배포에서, 이기종성은 네 가지 축을 따라 나타납니다: (i) 기록 도달 범위(체인을 얼마나 거슬러 올라가 스캔해야 하는지), (ii) 집계 너비(주소/포지션), (iii) 증명 주기(스왑당, 시간별, 일별 주기), (iv) 지연 시간 SLO(서비스 수준 목표). 그 차이를 고려해 보십시오: VIP 자격을 확인하는 DEX 훅은 단일 주소에 대해 30 일의 기록을 스캔하며 2-3 초 이내에 증명을 생성해야 합니다. 반면, 프로토콜 전체의 인센티브 분배는 매월 100,000 개의 주소를 처리하며, 지연 시간보다 처리량을 우선시합니다. 첫 번째 워크로드는 빠른 메모리 접근이 가능한 저지연 GPU 를 요구합니다; 두 번째 워크로드는 병렬 배치 처리에 최적화된 CPU 클러스터로부터 이점을 얻습니다. 단일 하드웨어 구성이 두 가지 모두를 효율적으로 처리하지 못합니다. 단일 "만능" 증명자 스택은 필연적으로 성능을 희생하게 됩니다. 뛰어난 FFT 처리량을 가진 STARK 우선 파이프라인은 기록 증명에서 뛰어날 수 있지만, 특화된 타원 곡선 사전 컴파일을 갖춘 SNARK 우선 스택은 거의 실시간 사용 사례에서 빛을 발할 수 있습니다.

확장하기 위해, Brevis 는 증명 컴퓨팅을 위한 '공개, 탈중앙화 마켓플레이스'를 옹호합니다. 요청자는 명시적인 제약 조건(마감 기한, 최대 수수료, 신뢰도 등급)과 함께 유형화된 작업을 게시하고, 증명자는 이를 실행하기 위해 입찰합니다. 가격 발견은 경매를 통해 이루어집니다; 정확성은 온체인 검증을 통해 강제됩니다; 그리고 서비스 품질은 경제적 인센티브(스테이킹, 평판, 그리고 시스템적으로 중요한 작업을 위한 이중화)에서 나옵니다. 이 구조는 클라우드 마켓플레이스를 반영하지만, 신뢰할 수 있는 청구 원장과 독점적인 대시보드를 제거하는 암호학적 강제성을 가집니다.

1.5. 기존 접근 방식의 한계

초기의 서비스형 증명 제공 및 클러스터 관리자는 가치를 입증하지만, 일반적으로 두 가지 구조적 한계에 직면합니다:

(1) **좁은 워크로드 지원.** 많은 스택이 하나의 증명 시스템, 하나의 VM, 또는 하나의 서킷 계열에 최적화되어 있습니다. 그들은 틈새 시장에서는 뛰어나지만, 다른 산술, 메모리 풋프린트, 또는 집계 전략을 가진 워크로드를 원활하게 흡수하지 못합니다. 작업 유형이 변경될 때(예: 작은 범위의 기록 스캔에서 재귀를 사용한 전체 zkVM 트레이스로), 시스템은 성능이 저하되거나 요청을 다른 사일로로 넘깁니다.

(2) **동종 노드 및 모놀리식 작업.** 유사한 노드를 가정하는 스케줄러는 이기종 하드웨어 전반에 걸쳐 작업을 분해하고 라우팅하는 데 어려움을 겪습니다. 실제로, 작업을 하위 작업(예: 다중 GPU 샤딩/파이프라이닝, 부분 증명 생성, 또는 병렬 데이터 수집)으로 나누는 것은 필수적이지만, 독립적인 운영자가 신뢰 없이 협력할 수 있도록 프로토콜 수준의 인터페이스와 인센티브가 필요합니다.

이러한 제약은 실제 애플리케이션의 처리량을 제한하고 증명까지의 시간을 저하시킵니다. 또한 특화된 용량(예를 들어, Keccak-heavy 트레이스에는 약하지만 다항식 커밋에 뛰어난 팜)을 숨겨 시장 효율성을 저해합니다.

1.6. 새로운 시장 설계: 진실된 온라인 이중 경매(TODA)

이러한 격차를 해소하기 위해, Brevis 는 진실된 온라인 이중 경매(TODA)의 속성을 가진 시장 메커니즘을 제안하고 구현합니다. 이중 경매는 많은 구매자와 많은 판매자가 있는 시장을 동시에 청산합니다. "온라인"은 작업과 용량이 시간 경과에 따라 도착함을 의미합니다. TODA 는 경매 라운드를 통해 증명 작업을 증명자와 매칭하며, 여기서 요청자는 최대 지불 의향을 입찰하고 증명자는 비용을 입찰합니다. "진실성"(인센티브 호환성)은 참여자가 자신의 진정한 가치(구매자) 또는 비용(판매자)을 입찰함으로써 유틸리티를 극대화하도록 보장합니다.

바람직한 속성은 다음과 같습니다:

- **인센티브 호환성(진실성):** 최선의 전략은 정직한 입찰입니다; 전략적 게임을 줄입니다.
- **개별 합리성:** 어떤 당사자도 손해를 강요받지 않습니다; 승자는 최대 자신의 입찰가(구매자)를 지불하거나 최소 자신의 요구가(판매자)를 받습니다.
- **예산 균형:** 메커니즘이 지속적인 보조금을 필요로 하지 않습니다; 수수료가 지불금을 충당합니다.
- **계산 효율성 및 낮은 지연 시간:** 경매는 SLA(서비스 수준 협약) 목표를 준수하기 위해 제한된 시간 내에 청산됩니다.

Brevis 내에서, 작업 유형(zkVM 실행, 재귀적 집계, 과거 데이터 증명, 래핑/필드 변환)은 일급 시민입니다. 증명자는 역량 프로필(하드웨어, 지원 시스템, 지연 시간 백분위수)을 광고합니다. 매칭은 스테이크 가중, 평판 인식 스코어링을 사용하며, 중요한 작업을 위해 선택적 이중화를 제공합니다. 이 조합은 이기종 노드가 분해된 작업에 대해 협력하고 범주 전반에서 공정하게 경쟁할 수 있게 합니다.

1.7. 성능 컨텍스트: Pico Prism 및 실시간 증명

단일 작업 지연 시간이 줄어들수록 컴퓨팅 마켓플레이스의 실행 가능성은 향상됩니다. Brevis 의 'Pico Prism'은 zkVM 증명을 위한 다중 GPU, 분산 아키텍처를 시연하며, 이는 기존 기술 수준을 실질적으로 향상시킵니다: 4,500 만 가스 상한을 가진 현재 이더리움 L1 테스트에서, 99.6%의 블록이 12 초 이내에 증명되었고 96.8%가 10 초 이내(이더리움 재단의 실시간 기준)에 증명되었으며, 64 개의 RTX 5090 클러스터에서 평균 약 6.9 초가 소요되었습니다.[1]

1.8. 시장 인프라로서의 토큰: BREV 의 역할

마켓플레이스는 결제 자금과 담보가 필요합니다. BREV 토큰은 세 가지 운영 역할을 수행합니다(섹션 5 에 자세히 설명됨). 첫째, 지불 매체입니다: Brevis ProverNet 내의 모든 작업 수수료, 검증, 결제는 BREV 로 표시됩니다. 둘째, 스테이킹 담보로 기능합니다: 증명자는 경매에 참여하고 서비스 수준 목표를 보장하기 위해 BREV 를 스테이킹합니다; 잘못된 행동이나 마감 기한을 놓치면 슬래싱을 통해 페널티를 받습니다. 셋째, 증명 네트워크가 특화된 롤업으로 실행됨에 따라, BREV 는 또한 네트워크 내 트랜잭션(작업 제출, 입찰, 증명, 스테이킹 업데이트)을 위한 가스 토큰 역할을 합니다. 이 세 가지 역할 설계는 컴퓨팅 및 블록스페이스에 대한 실제 수요를 BREV 에 대한 지속적인 수요와 연결하는 동시에, 신뢰성을 경제적으로 강제되는 서비스 속성으로 변환합니다.

1.9. 확장된 동기: 사용자가 실제로 원하는 것

최종 사용자의 관점에서 "블록체인 UX"는 '빠르고, 저렴하며, 공정한' 상호작용에 관한 것입니다. 트레이더는 스왑이 확정되기 '전에' 수수료 할인이 적용되기를 원합니다; 대출자는 제때에 지급되고 투명하며 감사 가능한 규칙에 따라 계산된 보상을 원합니다; 지갑은 개인 데이터를 유출하지 않고 계정 통찰력을 표출하기를 원합니다. 이러한 경험 중 어느 것도 모든 검증자가 온체인에서 수 초가 걸리는 분석 쿼리를 재생할 것을 요구하지 않습니다. 그것들은 쿼리가 올바르게 실행되었다는 '변조 방지 인증서'를 요구합니다.

UX 기본 요소로서의 상수 시간 검증.

이 백서에서 반복되는 주제는 상수 시간 검증이 UX 구성 요소가 된다는 것입니다: 일단 애플리케이션이 비즈니스 규칙을 검증 가능한 계산으로 표현하면, 체인은 데이터셋 크기에 관계없이 작고 예측 가능한 가스로 해당 규칙을 강제할 수 있습니다. 이는 이전에 "엄청나게 비쌌던" 아이디어(개인화된 수수료, 공정성 보장이 있는 스냅샷 기반 에어드랍, 또는 다중 체인 평판)를 일상적인 패턴으로 바꿉니다.

1.10. 실행 예: VIP 트레이더 프로그램

논의의 기초를 다지기 위해, DEX 풀의 VIP 프로그램을 고려해 보겠습니다. 목표는 트레이더가 해당 특정 풀(또는 여러 풀 집합)에서의 지난 30 일간 거래량, 그리고 추가적인 거버넌스 토큰 잔액에 대한 토큰 보유자 할인을 기반으로 스왑에 수수료 등급을 동적으로 적용하는 것입니다.

간단한 구현은 주소당 수천 개의 과거 이벤트를 읽고, 거래량을 집계하고, 멤버십 등급을 확인하고, 트랜잭션의 실행 창 내에서 결과를 사용할 수 있게 해야 합니다. 직접적인 온체인 구현은 불가능합니다: 이러한 기록을 저장하고 재계산하는 것은 엄청나게 높은 가스 비용을 발생시키며, 암호학적 증명 없는 오프체인 서버는 신뢰 가정을 다시 도입합니다.

ZK 코프로세서 아키텍처 하에서, 훅 컨트랙트는 작고 결정론적으로 유지됩니다. 코프로세서는 오프체인에서 무거운 계산을 수행하고, 증명을 생성하며, 훅에 상수 시간 내에 검증될 수 있는 간결한 증명 데이터 제공합니다.

1.11. 엔지니어링에서 경제학으로

엔지니어링 측면은 묻습니다: 우리가 증명을 충분히 빨리 생성할 수 있는가? Pico Prism 과 Brevis 프로덕션 시스템의 결과는 많은 종류의 워크로드에 대해 우리가 할 수 있음을 나타냈습니다.
경제학 측면은 묻습니다: 우리가 올바른 하드웨어를 올바른 작업에, 올바른 가격으로, 마감 기한 내에, 단일 운영자를 신뢰하지 않고 할당할 수 있는가? 이 백서의 나머지 부분은 스테이킹 기반 평판을 갖춘 온라인 이중 경매가 이러한 할당을 효율적으로 달성할 수 있다고 주장합니다.

1.12. 개요

이 백서의 나머지 부분은 다음과 같이 진행됩니다.

섹션 2 는 zkVM 아키텍처와 재귀적 구성을 강조하며 ZK 배경과 최근 진행 상황을 다룹니다.

섹션 3 은 Brevis 의 다양한 워크로드와 출시된 파트너십을 조사합니다.

섹션 4 는 이중 경매 체계에 대한 공식적인 설명을 제공합니다.

섹션 5 는 BREV 의 토큰 유틸리티와 가치 축적을 공식화합니다.

2. 배경

2.1. ZK 의 최근 진행 상황 (2024–2025)

영지식 증명 기술의 발전은 여러 전선에서 극적으로 가속화되었습니다. zkSNARK 시스템은 짧은 증명과 빠른 검증을 제공하며, 종종 강력한 엔지니어링 생태계를 갖추고 있지만 때때로 신뢰할 수 있는 설정이 필요합니다. zkSTARK 는 투명성(신뢰할 수 있는 설정 없음)과 양자 내성 보안 가정을 제공하며, 증명 크기는 더 크지만 FFT 및 해시 중심의 계산을 통한 뛰어난 병렬성을 가집니다. 고성능 STARK 기반 시스템을 위한 모듈식 툴킷인 Polygon 의 Plonky3 는 외부 감사와 채택 증가로 최근의 진전을 예시합니다. Nova 와 같은 재귀적 프레임워크는 폴딩 스킴을 사용하여 증명 구성 및 집계를 가능하게 합니다.

zkVM 은 중요한 증명 기본 요소로 부상했습니다. zkVM 은 머신 모델(종종 RISC-V 또는 WASM)의 올바른 실행을 증명합니다. 프로그램은 해당 ISA 로 컴파일된 다음, 각 명령어 단계를

zkVM 제약 조건으로 추가 변환하여 결과 증명이 전체 실행을 인증하도록 합니다. zkVM 은 유연성을 위해 최고 성능을 희생합니다: 각 애플리케이션에 대해 수동으로 서킷을 작성하는 대신, 개발자는 일반 VM 을 대상으로 하고 증명자가 최적화하도록 의존합니다.

Metric	SP1 Hypercube	Pico Prism	Improvement
RTP (<10s) coverage (36M gas limit)	40.9%	98.9%	2.4x higher coverage
RTP (<10s) coverage (45M gas limit)	N/A	96.8%	First to achieve
GPU cost (MSRP-based)	256K	128K	50% Reduction
Average proving time	10.3sec	6.04sec	71% Faster

그림 1. 2025 년 10 월 기준 "실시간 증명" 결과 비교

2.2. 실시간 이더리움 증명

2024-2025 년의 결정적인 도전 과제는 실시간 이더리움 증명이 되었습니다: 탈중앙화를 희생하지 않으면서 네이티브 롤업 보안과 베이스 레이어 검증을 가능하게 할 만큼 충분히 빠르게 L1 블록에 대한 유효성 증명을 생성하는 것입니다. 이더리움 재단의 벤치마크는 10 만 달러 미만의 하드웨어로 99%의 블록을 10 초 이내에 증명할 것을 요구했습니다.

Brevis 는 2025 년 10 월 'Pico Prism'을 발표하고 업계 최고 수준의 커버리지를 보고했습니다: 64 개의 RTX 5090 GPU 를 사용하는 소비자 등급 하드웨어에서 현재 4,500 만 가스 L1 이더리움 블록의 99.6%가 '12 초 이내'에 증명되었고, 96.8%가 '10 초 이내'(이더리움 재단의 "실시간" 기준)에 증명되었으며, 평균 증명 시간은 '약 6.9 초'였습니다. Brevis 팀은 또한 주어진 커버리지 목표에 대해 가장 가까운 경쟁자 대비 50%의 하드웨어 비용 절감을 보고하여, '달러당 3.4 배'의 성능 향상을 달성했습니다.[1] 미디어의 독립적인 보도 자료는 이러한 지표를 요약하며 '실시간 증명'이 베이스 레이어 검증을 위한 중요한 임계점을 넘었다고 평가합니다.

2.3. Brevis Pico 및 Pico Prism: 아키텍처 컨텍스트

Pico. Pico 는 Brevis 의 오픈소스 zkVM 으로, "글루-앤-코프로세서" 아키텍처를 채택합니다: 최소한의 고성능 코어와 무거운 암호학적 연산 및 도메인 특화 가속기를 위한 플러그형 코프로세서입니다.[2] 이 설계는 작업이 사용 가능한 특화된 가속 경로를 선택할 수 있게 하면서 프로그래밍 가능성을 보존하는 것을 목표로 합니다.

Pico Prism. Pico Prism 은 분산된 다중 GPU 병렬 처리 및 스케줄링으로 Pico 를 확장합니다. 개념적으로, Prism 은 실행 트레이스를 GPU 전반에 걸쳐 샤딩하거나 파이프라이닝하고 부분 결과를 최종 증명으로 집계하여, 메모리 압박과 처리량의 균형을 맞춥니다. 보고된 결과는 단일 실행 기록보다는 '지연 시간 제약 하의 커버리지'(< 12 초 / < 10 초)를 강조합니다. 이전 기술과의 비교는 1 을 참조하십시오. 자세한 내용은 [1]에서 찾을 수 있습니다.

시사점. 실시간 증명을 경제적으로 생성할 수 있다면, 베이스 레이어 검증은 상태 전환에 대한 사회적 합의에서 대부분의 블록에 대한 암호학적 유효성으로 이동할 수 있습니다. 이는 더

가벼운 노드, 롤업을 위한 더 빠른 완결성, 그리고 크로스체인 검증을 위한 새로운 보안 모델을 가능하게 합니다.

2.4. 네트워크화된 (시장) 접근 방식이 여전히 중요한 이유

더 빠른 zkVM 에도 불구하고, 워크로드 분포는 편향되어 있습니다: 어떤 작업은 거대하고(전체 L1 블록), 다른 작업은 지연 시간에 민감하며(브릿지 업데이트, 경매), 일부는 최선의 노력을 다하는 백그라운드 작업(과거 집계)입니다. 하드웨어 가용성도 마찬가지로 이기종입니다: GPU 가 하나인 단일 워크스테이션은 64-GPU 장비와 같지 않으며, 특화된 코프로세서는 노드의 하위 집합에만 존재할 수 있습니다.

유형화된 주문을 갖춘 양면 시장(작업 ↔ 증명자)은 매칭 문제를 해결합니다:

- **유형화된 작업:** zkVM 실행, 재귀적 집계, 래핑/압축, 필드 변환, STARK ↔ SNARK 래핑 등.
- **제약 조건:** 최대 수수료, 마감 기한, 필수 커버리지 백분위수(예: 95%의 인스턴스에 대해 10 초 이내 증명), 최소 하드웨어 프로필.
- **신호:** 평판, 과거 지연 시간 분포, 부하 시 신뢰성.

Brevis 의 경매 설계(유형화된 주문을 갖춘 진실되고 거의 최적에 가까운 이중 경매)는 양측의 인센티브 호환성을 유지하면서 이 시장을 청산하는 것을 목표로 합니다. 네트워크는 각 라운드마다 주문을 일괄 처리하고, 할당을 계산하며, 온체인에서 결제합니다; 충족되지 않은 사용자는 환불받습니다. (전체 공식적인 처리는 백서 뒷부분에 나옵니다.)

3. 다양한 워크로드

주요 DeFi 프로토콜, L2 생태계, 지갑 제공업체 전반의 프로덕션 배포는 Brevis 인프라가 처리하는 증명 워크로드의 스펙트럼을 보여줍니다. 이 섹션은 출시된 파트너십과 그들이 실행하는 구체적인 작업을 조사하여, 워크로드 이기종성이 어떻게 마켓플레이스 아키텍처를 필요로 하는지 설명합니다.

3.1. DEX 훅: 개인화된 수수료 및 데이터 기반 트레이딩

PancakeSwap Infinity (2025 년 5 월 출시)는 (i) 시간 가중 보유량에 따라 최대 45%의 수수료 할인을 적용하는 토큰 보유자 할인 훅과 (ii) 트레이더의 최근 풀별 거래량에 따라 수수료 등급을 매기는 거래량 할인 훅을 포함하여, 수수료를 사용자 맞춤형으로 적용하는 Brevis 기반 훅을 출시했습니다. 이러한 기능은 과거 이벤트를 스캔하고, 오프체인에서 통계를 계산한 다음, 라우터가 스왑 시점에 올바른 수수료 등급을 적용할 수 있도록 ZK 증명을 통해 온체인에서 결과를 검증해야 합니다. 초기 텔레메트리 보고: 6,151 건의 앱 요청 처리, 17,277,767 개의 증명 생성, 3,678 개의 고유 주소.

Uniswap v4 훅 (참조 디자인; 배포된 예제)은 주소당 지난 30 일간의 거래량을 완전히 오프체인에서 계산하고 이를 훅 컨트랙트에 다시 증명하는 "VIP 트레이더" 수수료 등급을

시연했습니다.[5] 원본 기사는 왜 DEX 가 모든 과거 거래를 온체인에 경제적으로 저장/쿼리할 수 없는지, 그리고 ZK 코프로세서가 어떻게 훅이 단순함을 유지하면서도 풍부하고 검증 가능한 기록의 이점을 누릴 수 있게 하는지 자세히 설명합니다. 동일한 패턴이 LP 로열티 보상, 변동성 인식 수수료, ZK-ML 지원 능동적 유동성 관리로 일반화됩니다.

QuickSwap 동적 수수료 (출시됨)는 Soneium(소니의 EVM L2)에서 WBTC/WETH 및 USDC/WETH 와 같은 풀을 시작으로 Brevis 기반 동적 수수료 훅을 도입했습니다. 트레이더는 (Brevis 에 의해 증명된) 지난 30 일간의 거래량에 따라 VIP 등급이 할당되고 할인된 수수료를 받으며, 추가적인 훅 변형(보유량 기반 및 변동성 기반)이 파이프라인에 있습니다.

3.2. Uniswap v4 라우터 가스 리베이트 (출시됨)

Uniswap v4 는 훅이 수수료 로직과 실행 보증을 수정할 수 있는 이기종 풀을 도입했습니다. 이 이기종성은 라우터가 다양한 풀 동작을 감지하고 처리해야 하므로, 애그리게이터(예: 1inch 및 ParaSwap)의 통합 비용을 높입니다. Uniswap 재단은 Brevis 와 협력하여 훅 지원 v4 풀을 통해 스왑을 라우팅하는 라우터에게 무신뢰 가스 리베이트를 출시하여, 채택을 가속화하기 위한 직접적인 경제적 인센티브를 창출했습니다.

리베이트 시스템은 Brevis 의 ZK 데이터 코프로세서를 통해 완전히 무신뢰로 작동합니다. 라우터는 적격한 트랜잭션 해시 목록을 제출하고 요청 ID 를 받습니다. Brevis 서비스는 영수증을 가져오고, 적격한 풀 ID 로 스왑을 필터링하며, 온체인 청구자 데이터를 통합하고, ZK 증명을 생성합니다. 일단 완료되면, 라우터는 요청 ID 로 쿼리하여 온체인 청구를 위한 증명과 콜데이터를 검색합니다.

온체인 청구는 리베이트 컨트랙트의 claimWithZkProof 를 통해 발생합니다. 이 함수는 다음을 강제합니다: (i) ZK 증명이 유효함, (ii) 지정된 블록 범위의 스왑이

Partner	Workload	Cadence/Scale
PancakeSwap	VIP/holding fee discounts via hooks	17.3M proofs; thousands of addrs
Uniswap v4	VIP tiers; LP loyalty; ZK-ML patterns	per-user periodic updates
QuickSwap	Dynamic fees (volume-based)	before-swap VIP checks
Euler	Incentra rewards on Arbitrum	4h epochs; 100k rEUL
Linea Ignition	1B LINEA program (Etherex/Aave/Euler)	12.1M proofs; 61.9k addrs
MetaMask	2.4% APR via Aave (Linea)	4h balance proofs
Uniswap v4 Rebates	Trustless gas rebates for v4 hooks	router initiated claims;

표 1. 파트너 및 도메인 전반의 대표적인 출시 워크로드.

이미 리베이트되지 않았음, (iii) 호출자가 등록된 청구자 주소임. 확인이 통과되면, ETH 리베이트가 지정된 수신자 주소로 지급됩니다. 라우터 정책에 따라, 리베이트는 운영 비용을 상쇄하거나, 사용자 수수료를 줄이거나, 재무부에 적립되어, 중앙화된 보상 원장이나 불투명한 계산 없이 v4 유동성과 채택을 가속화할 수 있습니다.

3.3. 대규모의 무신뢰 인센티브: Euler, Linea, MetaMask

Euler (Arbitrum 에서 출시됨)는 Brevis 의 무신뢰 인센티브 플랫폼인 Incentra 의 론칭

파트너였습니다. 2025 년 6 월 Euler 는 USDC, WETH, USDT, WBTC 볼트 전반에 걸쳐 '10 만 달러' 상당의 rEUL 을 분배하는 네 가지 대출 보상 캠페인을 실행했으며, 보상은 시간 가중 공급 잔액으로부터 4 시간마다 계산되고 온체인에서 검증되었습니다. 이 프로그램은 제로 커스텀 컨트랙트나 백오피스 스크립트 없이도 몇 주 이내에 유동성 깊이를 개선하고 차입 스프레드를 좁혔습니다; Incentra 가 데이터 수집, 증명, 온체인 분배를 처리했습니다.

Linea Ignition (출시됨)은 Etherex, Aave, Euler 전반에 걸쳐 10 억 개의 LINEA 토큰을 분배하는 10 주간의 프로그램을 실행했으며, '모든' 보상 계산은 오프체인에서 수행되고 온체인에서 증명되었습니다. 보고된 텔레메트리: 12,147,200 개의 증명과 61,902 개의 고유 주소.

Linea 의 MetaMask (출시됨)는 Brevis 와의 파트너십을 통해, MetaMask 카드 사용자에게 Linea 의 Aave 를 통한 USDC 대출/차입에 대해 2.4% 고정 APR 을 제공하는 프로그램을 지원합니다. Brevis 는 4 시간마다 시간 가중 잔액을 계산하고 자격을 증명합니다; 사용자는 Incentra 를 통해 청구합니다.

3.4. 생태계 확장: Linea, TAC, 그리고 기타

Linea (전략적): Ignition 외에도, Linea 는 디앱이 과거 데이터 접근, 복잡한 포지션 분석, 무신뢰 보상을 활용할 수 있도록 Brevis 를 일반적인 검증 가능한 컴퓨팅 레이어로 통합하여, 체인 수준의 채택을 보여줍니다.

TAC 및 Usual (출시됨): Brevis 는 TAC(텔레그램을 위한 EVM L1)에서 활성화되어 있으며, Usual 에 의한 Incentra 캠페인이 시간 가중 USD0++ 보유 및 유동성 활동에 보상합니다. 모든 것은 ZK 를 통해 증명되고 온체인에서 검증됩니다.

PADO (증명): 이전에, Brevis 는 PADO 사용자가 일정 기간 동안 자신의 가장 큰 Uniswap 거래를 증명하고, 온체인 신용 증명서로서 증명을 생성할 수 있게 했으며, 이는 여전히 과거 데이터 증명을 실행하는 가볍고 사용자 중심적인 워크로드입니다.

3.5. 네트워크 아키텍처에 대한 시사점

이러한 프로덕션 배포는 근본적으로 다른 계산 프로필에 걸쳐 있습니다. DEX 훅은 제한된 과거 범위를 스캔하는 트랜잭션별 확인을 위해 1 초 미만의 지연 시간을 요구합니다. 인센티브 분배는 수일에 걸친 주기 동안 수십만 개의 주소를 처리하며, 지연 시간보다 처리량을 우선시합니다. 크로스체인 증명은 재귀적 집계 파이프라인과는 다른 증명 시스템 최적화를 요구합니다. DEX 사용자의 30 일 거래량 계산은 SNARK 친화적인 타원 곡선 연산을 활용하는 반면, STARK 우선 시스템은 주기 기반 보상 분배에 필요한 FFT 중심의 집계에 탁월합니다.

단일 증명자 구성이 이 스펙트럼을 효율적으로 처리하지 못합니다. 저지연 GPU 증명에 최적화된 하드웨어는 CPU 에 바운드 배치 집계 작업에서 성능이 저하됩니다. 하나의 산술(Keccak 중심 트레이스 대 다항식 커밋 연산)에 최적화된 증명 시스템은 다른 워크로드 유형에 대해서는 성능을 희생하게 됩니다. (94,000 명 이상의 사용자를 위한 1 억 2,400 만 개의 증명에 걸쳐 검증된) 이 경험적 현실은, 모든 사용 사례를 처리하려는 모놀리식 인프라 대신

특화된 증명자가 유형화된 작업 범주 전반에서 경쟁하는, 섹션 4 에 자세히 설명된 마켓플레이스 아키텍처의 동기가 됩니다.

4. 이기종 증명자 네트워크를 위한 이중 경매

4.1. 시스템 모델

우리는 **이기종 증명 워크로드**를 지원하는 증명자 네트워크를 고려합니다. 예를 들어, 증명자 네트워크는 ZK 코프로세서 증명, zkVM 증명, 집계 증명 등의 생성을 지원할 수 있습니다. 일반성을 잃지 않고, 네트워크가 지원할 수 있는 K 가지 유형의 증명 워크로드가 있다고 가정하고, K = { 1, 2, ..., K }를 모든 증명 워크로드 유형의 집합으로 나타냅니다.

증명자 네트워크에는 증명 요청자 집합 N = {1, 2, ..., N}과 증명자 집합 M = {1, 2, ..., M}이 있습니다. 각 요청자 i 는 dkj 가 요청 di 내에 요구되는 type-k 증명의 수인, 여러 다른 증명 작업 유형의 벡터 di = (d1, d2, ..., dKj)로 분해될 수 있는 증명 요청 di 를 가집니다. 우리는 또한 di 를 요청자 i 의 **수요 벡터**라고 부릅니다. 만약 요청자 i 가 type-k 증명을 요구하지 않는다면, dkj = 0 입니다. 각 요청은 수요 벡터 내의 모든 증명 작업이 완료되어야 하는 마감기한을 가집니다. 더욱이, 각 요청은 원자적입니다: 수요 벡터 내의 모든 증명 작업이 100% 이행되거나, 또는 그 요청은 실패로 간주됩니다. 요청의 부분적인 이행은 가치가 없습니다.

위의 이기종 모델은 동종의 증명 작업만을 가정하는 이전 문헌 [4][13]보다 더 넓은 범위의 현실적인 증명자 네트워크를 특징으로 합니다.

구체적으로, 이기종 모델은 다음과 같은 이점을 가집니다:

- 첫째, 이기종 모델은 ZK 애플리케이션의 다양성을 포착합니다. 최근 ZK 기술의 발전과 ZK 기반 디앱의 폭발적인 증가로, 증명 작업은 점점 더 이기종화되고 있습니다. Brevis ProverNet 은 Brevis ZK 코프로세서 [3]와 Brevis Pico zkVM [2]을 모두 사용하는 애플리케이션의 증명 작업을 처리합니다. 미래에는 증명자 네트워크가 다양한 ZK 프레임워크 내에서 개발된 여러 ZK 솔루션(다중 zkVM)의 증명 작업을 지원할 수도 있습니다.
- 둘째, 이기종 모델은 여러 개의 더 작은 증명 하위 작업으로 구성된 모든 복잡한 증명 작업(예: EVM-검증 가능한 zkVM 증명 생성)을 특징지을 수 있습니다. 이는 증명자 네트워크에서 **분산 증명 생성**을 효과적으로 가능하게 하고 더 세분화된 증명 워크로드 할당을 달성하여, 제한된 증명 역량을 가진 증명자의 참여를 장려합니다.
- 마지막으로, 이기종 모델은 네트워크 내 증명자의 다양성을 포착합니다. 증명자의 하드웨어 역량 차이와 증명 작업의 이기종 하드웨어 요구 사항으로 인해, 증명자는 자신이 수행하는 증명 작업 유형에 대한 선호를 가질 수 있습니다. 메모리가 많은 증명자는 Plonk 기반 증명 작업을 선호할 수 있으며(큰 증명 키를 미리 메모리에 로드해야 하므로), 메모리는 제한적이지만 CPU 가 빠른 증명자는 STARK 기반 증명 작업을 선호할 수 있습니다.

예: zkVM 을 위한 이기종 증명 워크로드

대규모 zkVM 프로그램을 위한 EVM-검증 가능 증명을 생성하는 요청을 고려해 보겠습니다. Brevis 의 Pico zkVM 프레임워크에서, 이 요청은 다음과 같이 여러 다른 유형의 증명 하위 작업으로 분해될 수 있습니다:

(1) 첫째, VM 프로그램은 여러 청크로 나뉘며, 각 청크에는 증명(청크 증명)이 할당됩니다. VM 프로그램이 32 개의 청크로 나뉜다고 가정합니다.

(2) 그런 다음, 32 개의 청크 증명 각각은 증명 크기를 압축하기 위해 재귀 단계를 거칩니다(압축 증명).

(3) 다음으로, 32 개의 압축된 청크 증명은 단일 루트 증명이 생성될 때까지 이진 트리 방식으로 병합됩니다(병합 증명). 이 예에서, 이진 트리는 32 개의 리프와 31 개의 중간 노드를 가지므로, 병합 증명의 수는 31 개입니다.

(4) 그런 다음, 병합된 루트 증명은 기본 Poseidon2 해시 함수에서 사용하는 필드로부터 BN254 로 래핑되어, Plonk 증명(필드-랩 증명)에서 네이티브하게 검증될 수 있습니다.

(5) 마지막으로, 래핑된 증명은 Plonk 시스템(예: gnark)을 통해 재귀되어 EVM-검증 가능 증명(Plonk-랩 증명)을 생성합니다.

이 예에서, 요청은 다섯 가지 유형의 증명 작업을 가지며, 수요 벡터는 d = (32, 32, 31, 1, 1)입니다.

4.2. 이중 경매 프레임워크

문헌에 많은 경매 모델이 존재하지만 (예: VCG 경매 [14], 조합 경매 [9, 10], 전액 지불 경매 [6]), 우리는 증명자 시장의 동태를 더 잘 포착하는 양면 경매 또는 **이중 경매** [11] 프레임워크를 고려합니다. 이 모델에서, (1) 다수의 증명 요청자는 자신의 요청이 일정 비용으로 100% 충족될 수 있도록 증명자 리소스에 입찰합니다; (2) 다수의 증명자는 해당 요청을 처리하고 보상을 획득할 기회를 얻기 위해 경쟁합니다.

경매는 일련의 라운드로 진행됩니다. 각 라운드의 시작 시점에, 요청자 i 는 Bri = (di, vi) 입찰을 제출하며, 여기서 vi 는 작업을 완료하는 데 대한 요청자의 보고된 가치 평가입니다 (요청자가 지불할 의사가 있는 최대 수수료). 각 증명자 j 또한 Bpj = (wj , cj) 입찰을 제출합니다. 여기서, wj = (w1j, w2j, ..., wKj)는 증명자 j 의 보고된 공급 벡터이며, 각 wkj 는 증명자가 처리하기를 희망하는 유형-k 증명 작업의 수를 나타냅니다 1. cj = (c1j, c2j, ..., cK)는 보고된 한계 비용 벡터이며, 각 ckj 는 하나의 유형-k 증명 작업을 처리하는 비용을 나타냅니다. 만약 증명자가 특정 증명 유형(가령 유형-k 증명)을 지원하지 않는다면, wkj = 0 이고 ckj = ∞ 입니다.

요청자 i 로부터의 보고된 가치 vi 는 그것의 실제 가치 평가 vi 와 다를 수 있습니다. 마찬가지로, 증명자의 보고된 공급 벡터 wj 와 한계 비용 벡터 cj 는 그것들의 실제 대응값 wj 와

cj 와 다를 수 있습니다.

마지막으로, 만약 증명 요청이 한 라운드에서 수락되면, 증명은 바로 그 동일한 라운드 내에 생성되어야 합니다. 다시 말해서, 증명 생성 마감기한은 라운드의 종료와 일치합니다.

4.3. 문제 공식화

우리는 다음 두 가지 문제를 해결하는 증명자 시장을 위한 메커니즘을 설계하는 것을 목표로 합니다.

(1) **(증명 요청 할당)** 메커니즘은 각 요청자의 입찰이 경매에서 낙찰되는지와 필요한 증명자 리소스가 할당되는지를 결정해야 합니다. 또한, 메커니즘은 각 증명자에게 할당되는 (다른 유형들의) 증명 작업의 수를 결정해야 합니다. 구체적으로, x = (x1, x2, ..., xN)를 요청자들에 대한 할당 벡터라고 하며, 여기서 xi ∈ {0, 1}은 요청자 i 가 경매에서 낙찰되었는지 여부를 나타냅니다. yi = (y1, y2, ..., yK)를 증명자 j 에 대한 할당 벡터라고 하며, 여기서 yk 는 증명자 j 에게 할당된 유형-k 증명 작업의 수를 나타내는 정수입니다.

(2) **(가격 책정)** 메커니즘은 각 요청자가 플랫폼(즉, 경매인)에게 보내는 지불금을 결정해야 합니다. 지불 벡터는 p = (p1, p2, ..., pN)로 표기됩니다. 또한, 메커니즘은 각 증명자가 플랫폼으로부터 받는 보상을 결정해야 합니다. 증명자 j 에 대한 보상 벡터는 rj = (r1, r2, ..., rK)로 표기되며, 여기서 rk 는 할당된 유형-k 작업을 이행한 것에 대해 증명자 j 에게 주어지는 보상입니다. 증명자 j 가 받는 총보상은 아래 수식에 의해 표기됩니다.

요청자 i 에 대한 효용 함수는

$$U_i^r = \tilde{v}_i x_i - p_i$$

증명자 j 에 대한 효용 함수는

$$U_j^p = \sum_{j \in \mathcal{K}} U_{jk}^p = \sum_{j \in \mathcal{K}} \left(r_j^k - c_j^k y_j^k \right).$$

사회 후생 함수를 로 정의합니다

$$SW(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{N}} v_i x_i - \sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_j^k y_j^k.$$

1 분석을 단순화하기 위해, 우리는 각 요청의 마감기한이 경매 라운드의 기간과 같다고 가정하며, 따라서 보고된 공급 벡터의 모든 작업은 해당 경매 라운드 내에 완료되어야 합니다.

우리의 목표는 다음 속성들을 가진 메커니즘을 설계하는 것입니다:

- **정직성:** 증명 요청자와 증명자 모두 부정직하게 입찰하여 이익을 얻을 수 없어야 합니다; 즉, 모든 요청자 i 에 대해, Ur(v˜i)≥ Ur(vi), ∀vi 이고, 모든 증명자 j 에 대해,

  $$U_j^p(\tilde{\boldsymbol{r}}_j,\tilde{\boldsymbol{c}}_j) \geq U_j^p(\boldsymbol{r}_j,\boldsymbol{c}_j),\ \forall \boldsymbol{r}_j,\boldsymbol{c}_j.$$

- **예산 균형:** 모든 증명 요청자가 지불한 총금액은 모든 증명자에게 지급된 총보상보다 적지 않습니다, 즉, $\sum_{i\in\mathcal{N}} p_i \geq \sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}} r_j^k$.
- **개별 합리성:** 증명 요청자와 증명자 모두 실제 가치 평가를 보고할 때 음이 아닌 효용을 가집니다, 즉, $U_j^p(\tilde{\boldsymbol{r}}_j,\tilde{\boldsymbol{c}}_j) \geq 0,\ \forall j \in \mathcal{M}$
- **계산 효율성:** 메커니즘은 다항 시간 안에 실행되어야 합니다.
- **점근적 최적성:** 메커니즘은 총증명자 공급이 요청자의 수요에 비해 점점 더 충분해짐에 따라 최대 사회 후생에 근접해야 합니다.

4.4. TODA: 이기종 증명자 네트워크를 위한 진실되고 점근적으로 최적인 이중 경매 메커니즘

이 섹션에서는 이기종 증명자 네트워크를 위한 진실되고 점근적으로 최적인 이중 경매 메커니즘인 TODA 를 소개합니다. 메커니즘 설계는 섹션 4.4.1 과 4.4.2 에 설명되어 있습니다. 우리는 섹션 4.4.3 에서 'TODA'가 진실성, 예산 균형, 개별 합리성, 계산 효율성, 그리고 점근적 최적성을 달성함을 증명할 것입니다.

4.4.1. 증명 할당 알고리즘

최적의 증명 요청 할당은 사회적 후생 (1)을 극대화해야 하며, 이는 다음 정수 프로그래밍(IP) 문제를 해결함으로써 얻을 수 있습니다:

$$
\begin{aligned}
max_{\boldsymbol{x},\boldsymbol{y}} \quad & SW(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{N}} v_i x_i - \sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_j^k y_j^k \\
s.t. \quad & \sum_{i\in\mathcal{N}} d_i^k x_i = \sum_{j\in\mathcal{M}} y_j^k, \quad \forall k \in \mathcal{K}, \\
& x_i \in \{0,1\}, \quad \forall i \in \mathcal{N}, \\
& y_j^k \in \{0,1,...,w_j^k\}, \quad \forall j \in \mathcal{M}, k \in \mathcal{K}.
\end{aligned} \tag{2}
$$

정리 4.1. 사회적 후생 극대화 문제 (2)는 NP-hard 입니다.

증명. 우리는 작업 유형이 하나만 있고(|K| = 1), 증명자가 하나만 있으며(|J| = 1), 이 유일한 증명자의 이 유일한 작업 유형에 대한 한계 비용 c 가 0 인 특별한 경우를 고려합니다. 이 경우, 문제 (2)는 다음과 같이 축소될 수 있습니다:

$$
\begin{aligned}
max_{\boldsymbol{x}} \quad & \sum_{i\in\mathcal{N}} v_i x_i \\
s.t. \quad & \sum_{i\in\mathcal{N}} d_i x_i \le w, \\
& x_i \in \{0,1\}, \quad \forall i \in \mathcal{N},
\end{aligned} \tag{3}
$$

여기서, di 는 (유일한 유형의) 작업에 대한 요청자 i 의 수요이고, w 는 해당 작업에 대한 (유일한) 증명자의 공급입니다. 분명히, 축소된 문제 (3)는 NP-hard 인 배낭 문제입니다 [12]. 결과적으로, 원래 문제 (2) 또한 NP-hard 입니다.

정수 프로그래밍 문제 (2)를 풀기 위한 간단한 시도는 그것의 선형 완화를 취하는 것입니다. 하지만, 선형 완화에 대한 해가 정수일 것이라는 보장은 없습니다. [7][8]에서 사용된 "패딩" 방법에서 영감을 받아, TODA 는 "유령 증명 요청자"(PPR)를 도입하며, 이는 무한한 예산과 수요 벡터 $\boldsymbol{d}_{PPR} = (d_{PPR}^1, ..., d_{PPR}^K)$을 가진 가상의 강력한 요청자이며, $d_{PPR}^k = \max_{j\in\mathcal{M}} w_j^k$ 은 유형-k 작업에 대한 PPR 의 수요입니다. 다시 말해서, PPR 의 수요 벡터는 임의의 단일 증명자의 공급 벡터를 충족시킬 수 있습니다.

PPR 의 도입으로, TODA 의 증명 할당 알고리즘은 두 단계로 나뉩니다.

(1 단계) 첫 번째 단계에서는, PPR 을 경매에 도입하여 원래 정수 프로그래밍 문제 (2)의 수정된 선형 완화 문제를 풉니다. PPR 은 무한한 예산을 가지고 있으므로, 그것의 수요는 항상 충족될 것이며, 따라서 선형 완화 문제는 다음과 같이 됩니다:

$$
\begin{aligned}
max_{\boldsymbol{x},\boldsymbol{y}} \quad & SW(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{N}} v_i x_i - \sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_j^k y_j^k \\
s.t. \quad & \sum_{i\in\mathcal{N}} d_i^k x_i + d_{PPR}^k = \sum_{j\in\mathcal{M}} y_j^k, \quad \forall k \in \mathcal{K}, \\
& 0 \le x_i \le 1, \quad \forall i \in \mathcal{N}, \\
& 0 \le y_j^k \le w_j^k, \quad \forall j \in \mathcal{M}, k \in \mathcal{K}.
\end{aligned}
\tag{4}
$$

위의 선형 프로그래밍 문제 (4)의 최적해를 (x', y')로 나타냅니다. 각 요청자 i 에 대해, 임계 가치 평가 ϕ_i 를 (다른 모든 입찰자의 보고된 가치 평가가 변경되지 않은 상태에서) 요청자 i 가 $x'_i > 0$ 으로 승리하는 데 필요한 최소 가치 평가(즉, 가격)로 정의합니다. 보고된 가치 평가가 $v_i \ge \phi_i$ 를 만족하는 요청자 집합을 N 로 나타냅니다. 즉, N 는 $x'_i > 0$ 이고 $v_i \ge \phi_i$ 인 요청자들의 집합입니다. 오직 N 에 속한 요청자들만이 다음 단계로 진행합니다.

(2 단계) 두 번째 단계에서는, N 에 속한 요청자들과 모든 증명자 M 만을 포함하는 새로운 선형 프로그래밍 문제를 풉니다:

$$
\begin{aligned}
max_{\boldsymbol{x},\boldsymbol{y}} \quad & SW(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{N}} v_i x_i - \sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_j^k y_j^k \\
s.t. \quad & \sum_{i\in\mathcal{N}^*} d_i^k x_i = \sum_{j\in\mathcal{M}} y_j^k, \quad \forall k \in \mathcal{K}, \\
& 0 \le x_i \le 1, \quad \forall i \in \mathcal{N}^*, \\
& 0 \le y_j^k \le w_j^k, \quad \forall j \in \mathcal{M}, k \in \mathcal{K}.
\end{aligned}
\tag{5}
$$

위의 선형 프로그래밍 문제 (5)의 최적해를 (x'', y'')로 둡니다.
TODA 에 의한 최종 할당은

$$
\begin{aligned}
\boldsymbol{y}^{\text{TODA}} &= \boldsymbol{y}'', \\
x_i^{\text{TODA}} &= \begin{cases} x_i'', & \text{if } i \in \mathcal{N}^* \\ 0, & \text{if } i \notin \mathcal{N}^* \end{cases}.
\end{aligned}
$$

[7][8]에서 사용된 패딩 방법과 유사한 분석을 따르면, TODA 에 의한 해가 실행 가능함을 보일

수 있습니다. 즉, (xTODA, yTODA) 해는 정수 해입니다.

4.4.2. 가격 책정 정책

TODA 하에서, 요청자 'i'가 플랫폼에 지불하는 금액은 i ∈ N 이면 임계 가격 ϕi 와 같고, i ∉ N 이면 0 과 같습니다:

$$p_i^{\text{TODA}} = \begin{cases} \phi_i, & \text{if } i \in \mathcal{N}^* \\ 0, & \text{if } i \notin \mathcal{N}^* \end{cases}. \tag{6}$$

증명자들의 경우, VCG 와 유사한 가격 책정 방식이 사용되며, 여기서 증명자 j 가 받는 보상은 증명자 j 의 참여로 인해 다른 모든 입찰자의 사회적 후생에 발생한 변화로 정의됩니다:

$$r_j^{\text{TODA}} = \sum_{k \in \mathcal{K}} y_j^k c_j^k + SW(\mathcal{N}^*, \mathcal{M}) - SW(\mathcal{N}^*, \mathcal{M} \backslash \{j\}),\ \forall j \in \mathcal{M}. \tag{7}$$

4.4.3. TODA 의 메커니즘 분석

우리는 TODA 가 이전 섹션에서 언급된 바람직한 경제적 속성들: (1) 진실성, (2) 예산 균형, (3) 개별 합리성, (4) 계산 효율성 및 (5) 점근적 최적성을 달성함을 보입니다. 그 증명들은 [7][8]에서 사용된 패딩 방법과 동일한 논리 라인을 따르므로, 본 백서에서는 간결성을 위해 생략합니다.

정리 4.2. TODA 는 증명자가 제공하는 공급이 증명 요청자의 수요에 비해 점점 더 충분해짐에 따라, 진실성, 개별 합리성, 예산 균형, 계산상 다루기 쉬움, 그리고 점근적 효율성을 달성합니다.

4.5. Brevis Chain 을 이용한 경매 구현

이 섹션에서는 경매를 조정하는 **탈중앙화 경매인/플랫폼으로서 Brevis Chain** 을 이용한 위 경매의 구현을 설명합니다. 그림 2 는 상위 수준의 흐름을 보여줍니다.

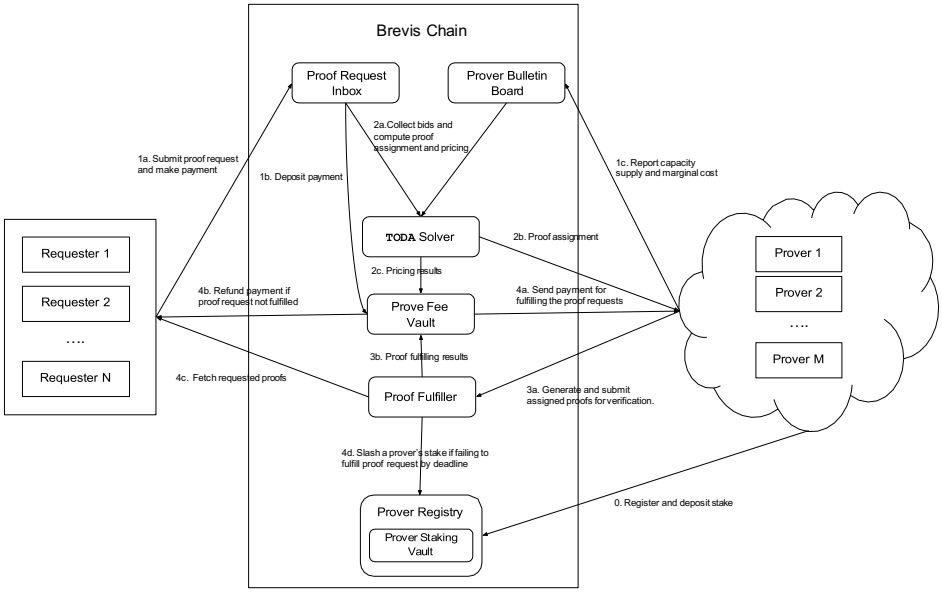

그림 2. Brevis Chain 을 이용한 증명자 네트워크 경매의 상위 수준 흐름

4.5.1 증명자 등록

증명 경매에 참여하기 전에, 증명자는 먼저 증명자 네트워크에 등록해야 합니다. 어떤 증명자든 무허가적으로 증명자 네트워크에 참여할 수 있습니다. 구체적으로, 각 증명자는 Brevis Chain 의 Prover Registry 컨트랙트에 등록하고 일정량의 스테이크를 예치해야 합니다. 만약 증명자가 증명 작업을 할당받았으나 마감 기한 전에 검증된 ZK 증명을 게시하지 못할 경우, 이 스테이크는 슬래싱될 것입니다.

4.5.2. 경매 라운드의 흐름

(1 단계) 경매 라운드가 시작될 때, 증명 요청과 증명자 용량 보고서를 수집하기 위한 입찰 수집 창이 있습니다. 각 요청자 i 는 Brevis Chain 의 Proof Request Inbox 컨트랙트에 증명 수요 벡터 di 를 보내고 Prove Fee Vault 에 (가치 평가 vi 와 동일한) 예치금을 납부할 수 있습니다. 동일한 창구 시간 동안, 각 증명자 j 는 자신의 공급 벡터 wj 와 한계 비용 벡터 cj 를 Prover Bulletin Board 에 보고해야 합니다.

(2 단계) 입찰 수집 창이 닫힌 후, TODA Solver 가 트리거되어 경매의 증명 할당과 가격 책정을 계산합니다 (섹션 4.4.1 및 4.4.2 에 설명된 메커니즘에 따라). 가격 책정 결과는 Prove Fee Vault 로 전달되고, 증명 할당 결과는 증명자들에게 전송됩니다. TODA Solver 는 Brevis Chain 이 지원하는 '네이티브 기본 요소'입니다.

(3 단계) 각 증명자는 TODA Solver 의 할당에 따라 증명을 생성하고, 저장 및 검증을 위해 Proof Fulfiller 에 증명을 제출합니다.

(4 단계) 증명이 성공적으로 검증되면, TODA Solver 가 생성한 가격 책정 결과에 따라

증명자에게 지불금이 전송되고, 증명 요청자는 필요한 증명을 가져갈 수 있습니다. 만약 증명자가 경매 라운드가 끝나기 전에 요청된 증명을 제출하거나 검증하지 못하면, 증명자의 스테이크는 슬래싱될 것입니다. 요청이 이행되지 않은 모든 요청자는 경매 라운드가 종료된 후 환불을 받게 됩니다.

위의 흐름은 Brevis Chain 에서의 경매 구현에 대한 상위 수준의 설명일 뿐이며, 증명 저장, 증명 요청자 및 증명자의 프라이버시 보존 등과 같은 많은 세부 사항을 생략하고 있습니다.

5. 토큰 유틸리티

5.1 개요

BREV 토큰은 Brevis 생태계의 핵심 유틸리티 토큰이다. BREV 는 증명 및 결제를 위한 지불 수단, 프로버 간 인센티브와 서비스 수준 보장을 정렬하는 담보 수단, 그리고 글로벌 시스템 파라미터를 설정하기 위한 거버넌스 토큰으로 기능한다. 본 절에서는 이러한 각 유틸리티를 설명하고, 이들이 어떻게 상호 결합되어 신뢰할 수 있고 시장 기반으로 가격이 형성되는 검증 가능한 계산 환경을 구성하는지를 설명한다.

5.2 검증 가능한 계산을 위한 지불 수단

Brevis ProverNet 과 관련된 모든 수수료는 BREV 로 지불된다. 여기에는 (i) zkVM 실행, ZK Data Coprocessor, ZKTLS Coprocessor 및 재귀적 집계(recursive aggregation)를 위한 증명 생성, (ii) 네트워크 상의 검증 및 결제, (iii) 결과 가용성 제공 및 영수증 게시와 같은 부가 서비스가 포함된다. 각 작업의 가격은 경매 메커니즘을 통해 결정된다.
또한 Brevis ProverNet 은 전용 롤업 형태로 배포될 수 있으며, 이 경우 BREV 는 네트워크 내 트랜잭션을 위한 가스 토큰으로 사용된다. 이를 통해 BREV 는 네트워크 사용에 따른 가스 비용 지불과 직접적으로 연결된 추가적인 유틸리티를 갖게 된다.

5.3 프로버의 작업 수주를 위한 스테이킹

분산된 네트워크에 참여하는 프로버는 작업을 할당받기 위해 반드시 BREV 를 스테이킹하거나, BREV 토큰 보유자로부터 위임된 스테이킹을 받아야 한다.
스테이킹은 다음과 같은 세 가지 역할을 수행한다.

(1) 시빌 공격 방지 및 접근 제어: 최소 유효 스테이킹 규모는 민감하거나 고가치 작업에 대한 접근을 제한하는 기준으로 작동한다.

(2) 경제적 정렬: SLA 미준수, 잘못된 증명 생성, 또는 이중 행위(equivocation)가 발생할 경우 스테이킹된 자산이 손실 위험에 노출된다.

(3) 처리 용량 신호: 충분한 성능 이력과 함께 더 큰 유효 스테이킹을 보유한 프로버일수록, 더 크거나 시간 민감도가 높은 작업에 대해 우선적으로 매칭된다.

증명을 요청하는 애플리케이션은 증명 요청과 함께, 해당 요청을 수락하는 프로버가 잠가야

하는 스테이킹 규모를 명시한다. 이는 사전에 정의된 검증 파라미터 내에서 요청이 제때 수행되지 않을 경우 발생할 수 있는 기회비용 또는 손실을 반영한다. 해당 스테이킹은 증명이 성공적으로 전달된 경우에만 해제된다.
토큰 보유자는 BREV 를 전문 프로버에게 위임할 수 있다. 이러한 위임 과정은 지분 위임 방식의 적극적인 참여 형태로 볼 수 있다. 위임된 토큰을 통해 프로버는 더 많은 증명 작업을 수주하고 추가 수익을 얻을 수 있으며, 그 대가로 증명 수수료의 일부를 위임자와 공유한다. 동시에 위임받은 프로버가 SLA 를 위반할 경우, 위임된 스테이킹 역시 함께 슬래싱되므로, 위임자는 다양한 프로버의 성과를 신중히 평가하고 위임 상태를 주기적으로 조정할 필요가 있다.

5.4 SLA 위반에 대한 슬래싱

Brevis 시스템 전반은 영지식 증명에 기반한다. 프로버가 생성한 잘못된 증명이 작업 결제 컨트랙트에 의해 성공적으로 검증되는 것은 불가능하다. 네트워크의 정상적인 운영을 저해할 수 있는 가장 큰 리스크는, 사전에 약속한 프로버가 서비스 수준 계약(SLA)을 위반하는 경우이다. 여기에는 마감 기한 미준수, 추가 보상 요구, 또는 사전에 합의된 수준보다 낮은 보안성이나 더 큰 크기의 증명을 생성하는 행위가 포함된다.
프로버가 SLA 를 위반할 경우, 증명 요청자는 스테이킹 컨트랙트를 통해 해당 프로버의 잠긴 스테이킹을 슬래싱할 수 있다. 슬래싱 비율은 시스템 파라미터로 정의되며, 초기값은 1%로 설정되어 있다. 이 비율은 향후 프로토콜 거버넌스 과정을 통해 점진적으로 조정될 수 있다.

5.5 프로토콜 파라미터에 대한 거버넌스

BREV 토큰은 온체인 거버넌스 프로세스를 통해 주요 시스템 파라미터를 설정하는 거버넌스 토큰으로도 사용된다. 초기 단계에서 거버넌스 대상이 되는 파라미터는 다음과 같다.
(1) 허용 가능한 증명 크기: Brevis ProverNet 은 1MB 미만의 증명만 허용한다.
(2) 보안 수준: Brevis ProverNet 은 모든 증명에 대해 100 비트 이상의 보안 수준을 요구한다.
(3) 슬래싱 비율: SLA 를 위반한 프로버에 대해 잠긴 스테이킹의 1%가 슬래싱된다.
(4) 경매 시장 수수료: 요청자가 지불한 수수료 중 3%가 Brevis ProverNet 에 할당된다.

6. 토크노믹스 (Tokenomics)

BREV 토큰은 Brevis 생태계의 핵심 유틸리티 및 가치 축적 자산으로서, 네트워크의 지속적인 성장, 참여 인센티브 제공, 그리고 프로버, 개발자, 사용자 간의 경제적 정렬을 보장하도록 설계되었다. BREV 의 총 발행량은 1,000,000,000 개로 고정되어 있으며, 토큰 배분 구조는 장기적인 생태계 건전성, 탈중앙화, 그리고 운영의 지속 가능성 간의 균형을 목표로 설계되었다.
BREV 토큰의 배분은 기여자, 사용자, 이해관계자 간의 인센티브를 정렬하도록 구성되어 있다.

- **생태계 개발 (37%)**: 생태계 성장, 연구 및 개발, 전략적 파트너십, 초기 마켓 메이킹, 그리고 장기적인 프로토콜 확장을 지원하기 위한 물량이다.
- **커뮤니티 인센티브 (28.7%)**: 프로버, 스테이커, 커뮤니티 기여자 및 개발자 통합에 대한 보상을 포함한다.

- **팀 (20%)**: Brevis 의 현재 및 향후 핵심 개발자와 기여자에게 배정된다.
- **투자자 (10.8%)**: Brevis 의 개발 및 출시를 지원한 시드 투자자에게 배정된다.
- **에어드롭 (3.5%)**: 자격 요건을 충족한 기여자 및 커뮤니티 구성원을 대상으로 한 초기 에어드롭 물량이다.

생태계 개발 물량과 커뮤니티 인센티브 물량은 TGE(Token Generation Event) 시점의 초기 언락 이후 24 개월에 걸쳐 선형적으로 베스팅되며, 출시 시점에 각각 전체 공급량의 14.50%와 7.50%가 유통된다. 에어드롭 물량 중 3%는 TGE 시점에 언락되며, 나머지 0.5%는 TGE 이후 6 개월 차에 언락된다.

팀 및 투자자 배정 물량은 초기 언락 없이 첫 1 년 동안 전량 락업되며, 이후 24 개월에 걸쳐 선형적으로 베스팅된다. TGE 시점의 전체 유통량은 총 공급량의 25%이다.

7. 비즈니스 모델

Brevis 의 비즈니스 모델은 **검증 가능한 계산(verifiable computation)**을 중심으로 한 지속 가능하고 사용량 기반의 경제 구조에 기반하고 있습니다. ZK Data Coprocessor, Pico zkVM, 그리고 향후 Ethereum 전체 블록 증명을 사용하는 애플리케이션을 포함하여 Brevis 위에 구축된 애플리케이션들은 지속적으로 영지식 증명 계산이 필요한 워크로드를 생성합니다. 이러한 워크로드는 계산 수요(order flow)를 형성하며, 이는 계산을 위한 탈중앙화된 마켓플레이스인 **Brevis Prover Network** 를 구동하는 기반이 됩니다.

이 마켓플레이스 내에서 독립적인 프로버 및 프로버 풀들은 애플리케이션이 제출한 계산 요청을 수행하기 위해 경쟁합니다. 각 증명 작업은 **서비스 수준 계약(SLA) 메커니즘**과 **TODA 이중 경매 시스템**에 의해 운영되는 투명한 경매 과정을 통해 매칭됩니다. 성능, 신뢰성, 그리고 스테이킹 규모가 작업과 보상이 어떻게 배분되는지를 결정합니다. 프로버는 네트워크에 참여하기 위해 BREV 토큰을 스테이킹해야 하며, 이는 참여에 대한 의지를 나타내는 동시에 서비스 품질을 보장하는 역할을 합니다. 더 많은 BREV 를 스테이킹하고 더 나은 SLA 이력을 보유한 프로버일수록 더 많은 증명 작업과 이에 상응하는 수익을 배분받게 됩니다.

이 모델은 자기 강화적인 가치 순환 구조를 형성합니다. 더 많은 애플리케이션이 Brevis 를 통합하고 검증 가능한 계산에 대한 수요가 증가함에 따라, 네트워크 사용료 지불과 스테이킹을 위해 필요한 BREV 에 대한 수요도 증가합니다. 그 결과 스테이킹은 네트워크의 신뢰성과 처리 용량을 강화하고, 이는 다시 더 많은 애플리케이션과 계산 워크로드를 유치하게 됩니다.

증명 워크로드가 증가함에 따라 Brevis Prover Network 는 전용 롤업으로 전환될 수 있으며, 이를 통해 경매와 결제의 확장성과 효율적인 조정을 보장할 수 있습니다. 이 롤업 내에서 이루어지는 모든 거래(작업 대금 지급, 스테이킹, 프로버 보상 포함)는 BREV 토큰으로 표시되며, 네트워크의 경제 흐름에 내재적인 유틸리티를 직접적으로 포함합니다. 또한 각

ProverNet 경매가 성공적으로 완료될 때마다 소액의 프로토콜 수수료가 Brevis 프로토콜 트레저리에 적립되어, 네트워크 활동에 따라 확장되는 반복적이고 투명한 수익원을 형성합니다.

이러한 구조를 통해 Brevis 는 계산 수요를 경제적 가치로 전환하며, 토큰 인센티브를 실제로 측정 가능한 네트워크 사용량 및 장기적인 생태계 성장과 정렬시킵니다.

8. 로드맵

2025 년 4 분기
• 서드파티 프로버(증명자)를 온보딩하며 **Brevis 프로버 네트워크** 테스트넷을 출시합니다.
• **Proving Grounds** 사용자 활성화 캠페인과 TGE(토큰 생성 이벤트) 이전 생태계 롤아웃을 완료합니다.
• **TGE** 를 2025 년 12 월 초에 목표로 합니다.

2026 년 상반기
• 16 대의 일반 소비자용 GPU 에서 **실시간 이더리움 블록 증명**을 완성합니다.
• 검증 가능한 웹 데이터와 AI 추론을 위해 **ZK-TLS 코프로세서**를 통합합니다.
• 분산형 프로버 작업 할당을 위한 **TODA 경매 메커니즘**을 배포합니다.
• **Intelligent DeFi**, **스테이블코인/RWA 인센티브 레일**, **ZK 기반 예측 오라클** 분야로의 채택을 확대합니다.

2026 년 하반기
• **Pico zkVM** 을 위한 **맞춤형 하드웨어 가속(FPGA / ASIC)**을 연구합니다.
• **Brevis 프로버 네트워크**를 전 세계 검증 가능한 컴퓨팅을 위한 분산형 마켓플레이스로 확장합니다.
• **Pico zkVM** 을 이더리움 L1 의 실시간 증명 스택에 포함시킵니다.
• DeFi, AI, 크로스체인 시스템에 검증 가능한 컴퓨팅을 통합하는 빌더들을 지원하기 위해 **Brevis 생태계 그랜트 프로그램**을 시작합니다.

9. 결론

이 백서는 간단한 명제를 제시합니다: 복제된 실행의 성능 및 비용 한계를 돌파하는 방법은 '검증 가능한 컴퓨팅'을 애플리케이션과 합의 사이의 기본 인터페이스로 만들고, 그 컴퓨팅을 '공개적이고, 이기종적인 마켓플레이스'를 통해 공급하는 것입니다. Brevis ProverNet 은 증명 생성을 수요가 '유형화된 작업'으로 도착하고 공급이 다양한 하드웨어 프로필과 증명 시스템 전문화를 가진 다양한 증명자 집합인 일급 시장으로 취급함으로써 이 아이디어를 운영합니다. 매칭은 명확한 서비스 수준 목표와 암호학적 결제를 갖춘 온체인 경매를 통해 발생하므로, 정확성과 지불은 단일 운영자에 대한 신뢰에 의존하지 않습니다.

참고 문헌

[1] Brevis pico prism: Real time proving, https://blog.brevis.network/2025/10/15/pico-prism-99-6-real-time-proving-for-45m-gas-ethereum-blocks-on-consumer-hardware/.

[2] Brevis pico zkvm, https://pico-docs.brevis.network/.

[3] Brevis zkcoprocessor, https://coprocessor-docs.brevis.network/.

[4] Succinct network: Prove the world's software, https://www.provewith.us/.

[5] Uniswap v4 hook: Brevis zk, https://blog.brevis.network/2023/11/01/ uniswap-v4-hook-brevis-zk-coprocessor-data-driven-dex-experiences/.

[6] M.R. Baye, D. Kovenock, and C.G. De Vries, The all-pay auction with complete informa- tion, Economic Theory 8 (1996), pp. 291–305.

[7] S. Chen, M. Liu, and X. Chen, A truthful double auction for two-sided heterogeneous mobile crowdsensing markets, Computer Communications 81 (2016), pp. 31–42.

[8] L.Y. Chu, Truthful bundle/multiunit double auctions, Management Science 55 (2009), pp. 1184–1198.

[9] S. De Vries and R.V. Vohra, Combinatorial auctions: A survey, INFORMS Journal on computing 15 (2003), pp. 284–309.

[10] M. Dong, G. Sun, X. Wang, and Q. Zhang, Combinatorial auction with time–frequency flexibility in cognitive radio networks, in 2012 Proceedings IEEE INFOCOM. IEEE, 2012, pp. 2282–2290.

[11] D. Friedman, The double auction market institution: A survey, in The double auction market, Routledge, 2018, pp. 3–26.

[12] R.M. Karp, Reducibility among combinatorial problems, in 50 Years of Integer Program- ming 1958–2008: from the Early Years to the State-of-the-Art, Springer, 2009, pp. 219–241.

[13] A. Lazzaretti, C. Papamanthou, and I. Hishon-Rezaizadeh, Robust double auctions for resource allocation, Cryptology ePrint Archive (2024).

[14] H.R. Varian and C. Harris, The vcg auction in theory and practice, American Economic Review 104 (2014), pp. 442-445.