

[트랙②] 블록체인 난제 해결 챌린지 기획서(심사위원 확인용)

서비스(아이디어) 제목		고속 영지식 증명 생성 및 실시간 검증 지원 프라이버시 보호 PoR 시스템		
팀명		Pororo	팀원	정권호, 김경빈, 김승우, 문명균, 윤건호
서비스(아이디어) 주요 개요		<p>현존하는 PoR 시스템은 증명 생성 관련 성능 부분과 실시간성 측면에서 단점이 존재. 본 팀에서는 기존 PoR 시스템의 느린 영지식 증명 생성 시간과 부족한 실시간성 문제를 해결하기 위해 Pedersen 약정과 약정 동반 SNARK를 결합한 새로운 고속 영지식 증명 생성 및 실시간 검증 지원 프라이버시 보호 PoR 시스템을 제안. 새롭게 제안한 PoR 시스템은 토큰 증권 시장에 적용하여 토큰 증권 보유 수량과 관련하여 효율적인 증명 생성 시간과 높은 실시간성을 가지는 토큰 증권 시장 서비스를 제공 가능함. 추가적으로 기존 PoR시스템과 같이 프라이버시를 보장함.</p>		
세 부 내 용	필요성 (25점)	<p>1. 블록체인 난제 정의 및 제안배경은 무엇인가?(기존, 최신 이슈 등 사례를 들어 작성)</p> <p>디지털 자산 관련한 회계 및 공시 측면에서도 2023년 6월 30일 가상자산법이 국회 본회의를 통과하면서 가상자산을 발행하거나 보유한 기업이 명확하고 상세한 정보를 공개하는 회계투명성을 요구. 법안의 내용 중 가상자산 사업자의 경우 고객위탁 가상자산을 보유한 경우 사업자의 재무제표에 자산 및 부채 인식 여부의 불분명성 존재. 즉 구체적인 기준이 명확하지는 않지만 디지털 자산의 보유상황 및 회계 정책에 대한 불분명성을 해결하고 검증가능성의 필요성이 대두. 그러나 투명하게 보유상황의 공개는 사용자들의 프라이버시 보호와 대립되는 상황 발생. 전술한 상황에 대한 해결책으로 수탁기관은 보유한 자산을 자발적으로 증명하는 PoR의 개념 등장. 가장 대표적인 Binance의 방식은 Merkle Tree와 zk-SNARK를 이용하여 각 사용자들의 정보를 보호함과 동시에 자산 보유 상황을 증명함. 하지만 상기 방식은 성능과 실시간성이라는 두 가지 측면에서 문제 존재. 성능적인 측면에서는 증명 생성 시간이 수십 시간이 소요됨. 또한 생성되는 증명은 스냅샷에 대한 증명이므로 낮은 실시간성을 가지고 있다고 볼 수 있음. 전술한 두 가지 측면의 문제를 해결하기 위해 고성능의 영지식 증명 생성 및 실시간 검증을 지원하는 프라이버시 보호 PoR 시스템을 제안함.</p>		
	구체성 및 차별성 (25점)	<p>2. 제안한 해결방안을 통해 블록체인 서비스의 어떠한 난제를 해결할 수 있는가?</p> <p>현존하는 PoR 시스템에서 존재하는 느린 PoR 증명 생성 시간과 낮은 실시간성 문제를 해결가능</p> <p>3. 기존 블록체인 난제 해결방안과 차이점은 무엇인가?</p> <p>영지식 증명 기술을 사용하는 것은 동일하지만 멤버십에 대한 검증을 위해 기존 방식은 Merkle Tree를 사용하지만 본 팀에서 제안하는 방식은 Pedersen 약정과 약정 동반 SNARK(Commit-carrying SNARK)를 사용하는 것이 차이점. 기존 방식은 멤버십에 대한 회로가 Tree 높이만큼의 해시 함수 연산이 포함되기 때문에 증명 생성 시간이 현저하게 느려짐. 이와 다르게 본 팀에서 제안하는 방식은 약정 동반 SNARK를 통해 증명값과 동시에 약정값이 생성되므로 약정값에 대한 회로가 없이 증명 생성 시간이 효율적임. 또한 느린 증명 생성 시간으로 인해 스냅샷 이후 증명을 생성하는 기존 방식 대비 제안한 방식은 트랜잭션마다 실시간으로 스마트 컨트랙트를 통해 반영되기 때문에 기존 대비 높은 실시간성을 지님.</p>		
	문제이해도 (25점)	<p>4. 정의한 블록체인 난제를 해결하기 위해 사용된 블록체인 기술은 무엇인가?</p> <p>암호학적 도구를 사용하여 난제 해결을 시도함. 주요 기반 기술을 아래와 같음</p> <ol style="list-style-type: none"> 1) Pedersen commitment: 약정 스킴(Scheme)의 한 예로 그룹 선형 연산으로 약정값을 생성함. 2) 약정 동반 SNARK(Commit-carrying SNARK): 약정 후 증명 SNARK(Commit-and-prove SNARK)의 소분류이며 키 생성 시 약정 키가 증명 키와 검증 키와 같이 생성됨. 또한 증명 과정 시 증명값과 약정값을 동시에 생성한다는 특징을 지님. <p>상기 언급된 두 암호학적 도구를 바탕으로 zk-SNARK의 대표적인 시스템인 Groth16은 ccGro16으로</p>		

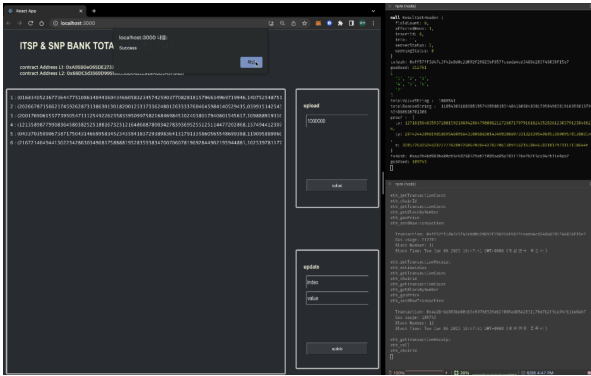
		<p>변환되는데 이 때 Groth16의 증명값 생성과정이 그룹에 대한 선형 연산이므로 생성되는 약정값은 Pedersen 약정값의 형태를 지님. 이를 이용하면 Pederse 약정 연산에 대한 회로없이 Pedersen 약정값을 고승으로 생성하고 증명 및 검증이 가능함.</p>
활용가능성 (25점)	<p>5. 제안한 블록체인 난제 해결방안이 실현 가능한가?</p> <p>암호학적 도구를 사용하기 위한 라이브러리 다수 존재(Rust, C++)하며 스마트 컨트랙트 내에서도 그룹 연산(BN256)을 제공하여 구현 가능. 서버에서 증명 생성 시 zk-SNARK를 위한 라이브러리(Arkworks, Libsnark)가 존재하여 실현 가능.</p> <p>6. 블록체인 난제 해결방안 적용시 기대효과와 파급력은 어떻게 되는가?</p> <p>현존하는 방식을 제안하는 방식의 PoR 시스템으로 변경한다면 느린 증명 생성 시간에 따른 성능 및 실시간성의 비효율성 개선 가능. 블록체인 기반의 디지털 자산 거래소나 금융 기관의 신뢰성을 제고하며 안전한 거래 환경 조성 실현. 특히 토큰 증권같은 디지털 자산 시장에서 프라이버시를 보호함과 동시에 거래소와 금융 기관 사이의 신뢰 관계 강화</p>	
프로토타입 설명 (가산점 10점)	<p>7. 정의된 블록체인을 해결하기 위해 어떤 기술을 적용하여 구현하였는가?</p> <p>본 팀의 아이디어에 대한 프로토타입 구현하기 위해 Javascript를 주요 언어로 , Solidity, Rust를 활용. 프레임워크의 경우 React 기반으로 Web3를 결합하여 전체적인 프로토콜 구현.</p> <ul style="list-style-type: none"> - 약정 동반 SNARK는 Rust로 구현된 LegoSNARK 모듈 사용. - 스마트 컨트랙트 내에서는 영지식 증명을 검증하기 위한 Pairing 연산에 대한 assembly 코드를 구현하였음. - Pedersen commitment의 경우 javascript 내 직접 구현 	

그림 1 PoR Demo

※ 요약서는 최대 2페이지 이내로 반드시 작성