

ZKU - Assignment 4

Email – stanleychiu@protonmail.com

Discord – HKerStanley#4125

GitHub – <https://github.com/HKerStanley/zk-uni> (./asset/week_4)

Question 0: Which stream do you choose (answer with A or B)

I am going for Stream B to build ZK Dapps

Question 1: Scaling the future

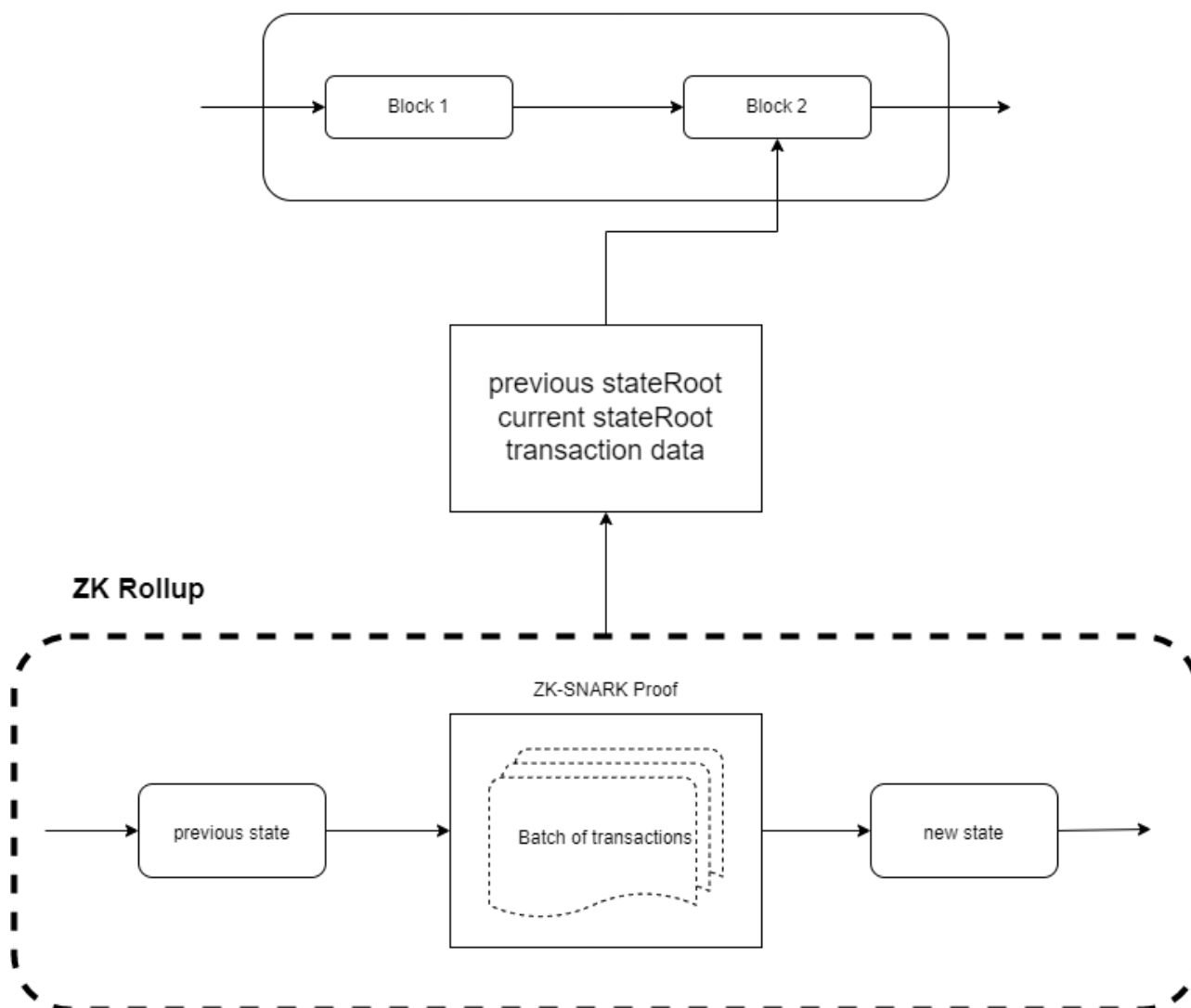
1. There are 4 different approaches for L2 solution to try to solve the scalability trilemma:

1. State Channels: As most of the transactions moved off-chain, L1 blockchain will only act as a settlement layer which massively increase the amount of transactions that can be handle while keeping the cost low and fast. However, the channel cannot transact with people outside the channel, which limit the users from complex application. Also, opening such channel requires user to lock up liquidity, which if it scales it can be capital inefficient. Also participants need to actively monitor the network to ensure the fund is secure, which can cost inefficiency and complexity.
2. Sidechains: Basically another blockchain that pegged to the L1 blockchain, it scales L1 blockchain by using a different consensus algorithm. However they typically made a trade-off in decentralization. And since they are independent blockchain and usually have their own security model, they can be less secure than the L1 blockchain for example Ethereum.
3. Plasma: Plasma works very similar to a sidechain, but instead of an independent blockchain from L1 they are a series of smart contracts. Merkle Tree will be constructed from transactions and the root will be committed on L1 blockchain so that it still enjoys the security by the L1 blockchain. However, since an operator is needed for posting the merkle root to L1 blockchain, they will have the power to perform data availability attack that withhold transaction data. Lack of data can lead to acceptance of invalid blocks as there will be no way to prove the validity.
4. Rollups: It works similar to Plasma but this time we also post the transaction data on-chain. The amount of data is keep as minimum amount required to locally validate the rollup transactions. Moving transactions to the rollup later and compress transaction data help to scale the L1 blockchain with same level of security. But since we are still posting data on-chain, the scalability is limited. And since rollup is working on a different layer, it makes liquidity fractured and lose the composability when building on L1 blockchain.

2. ZK-Rollup follows the "Don't trust, verify" mentality that it execute and validate transactions off-chain. It make sure the batch of transactions is correct and generate proof that can be used for verification. Both the proof and transaction data will be posted on chain so that anyone can verify the data if needed. The reason I am more lean on ZK rollup is about the latency. User experience is better

as the withdrawal of assets is faster in ZK rollup, while in Optimistic rollup user need to wait for a "challenge period" if there are fraud proof. Also, since transactions are posted after verification, ZK rollup is more secure than other rollup.

Layer 1



3. Stateless client is a concept that allow nodes to validate a block without storing a full Ethereum state. Instead, the state which is required for validating a block is delivered as a witness. From here we can apply zero knowledge concept that such witness is used to verify the next state without the need of revealing/storing the previous state.

Question 2: Roll the TX up

3. Major advantages of zkPorter is the exponential growing TPS and significant cost reduction. Also, in their Proof of Stake protocol, operators or guardians cannot steal fund but only freeze them, which is more secure. Generally speaking I think zkPorter is good for majority of users as speed and cost are their major concern. It is also providing an extra option to users when a transaction is not that sensitive to require best security. However, since zkPorter stores data off-chain and depends on validators' stake, it can be less secure compare to storing on-chain.

Question 3: Recursive SNARK's

1. Recursive SNARK is a scaling solution of a simple SNARK by recursion. As SNARK can verify and be verified by computations, such computation can be chained and lead to a fixed size of output. This solves the problem of a linear growth of time complexity for computing a proof for a large amount of input. One possible security drawback I can think of is that since all computation is based on the previous SNARK proof, it can be attacked by withholding the previous proof or somehow fake a previous proof. And since original data will not be available such malicious attack might be more difficult to be noticed.
2. Kimchi is a proof system in the Mina protocol and is an improved version of PLONK. They use a bulletproof-style polynomial commitment in Kimchi so there is no need for a trusted setup. Also, they added more registers to PLONK so it can take more input and perform more complex computation.

Question 4: Final Project Ideas

1.
 - Idea 1: Building the [Stratego](#) in zero knowledge gameplay. In the original gameplay, when a player attacks opponent's piece by moving to occupied space, both players have to reveal the piece. With zero knowledge movement we will not need to reveal both pieces to guarantee an attack result. At the same time we don't need a third person as the judge. I believe games are very good at promoting new technology to general public while improving the existing gameplay.
 - Idea 2: A zero-knowledge prover of citizenship or identity document. We are seeing a trend of traditional services like banking going digital/online. It's normal for these services to request users to upload their passport or ID card or sensitive information for Know Your Customer (KYC). The security concern is clear when users are being asked to pass such data online and I think privacy can be improved with zero knowledge proof in such process.
2. Zero knowledge is relatively new to public and there are very few fun products in the market. Building a fun board game in my point of view is a very efficient way to onboard new users. However, this simple game with extra technical complexity may not be very attractive to players. Compared to my second idea that already has a strong use case the game is not very valuable in solving real world problems. However, my second idea is very ambitious with quite a large technical scope which I am not confident to be able to finish on time. I am more prefer the game idea as I have game development background so it should be within my capability.

Question 5: Thinking in ZK

1. Question to Mina protocol, how is the consensus layer looking like in Mina blockchain? We are expecting a large amount of block producers and snark producers, how is the producer being chosen? How does the whole blockchain come to an consensus if the number of participants is so large and will it take a linear or even exponential growth of time to come to a consensus?