Google Colaboratory

Announcements

Project: final version due 5/1

Final Exam on Monday, May 8 at 9-11am -- cumulative test that covers all lectures (through April 25), recitation labs, homework assignments, and assigned reading

Please submit your course evaluation at https://bu.campuslabs.com/courseeval/

Office hours moved to Thursday 4/28/2023 after class.

Lecture 24: Zero-knowledge and Optimistic rollups

Lecture taken from the ethereum developer page and from Vitalik's blog

Recap:

The Lightning Network is a layer 2 scaling solution for Bitcoin that allows for faster and cheaper transactions by creating off-chain payment channels between users. It aims to increase the capacity and efficiency of the Bitcoin network while maintaining its security and decentralization.

Zcash and Mina are examples of blockchains that use zero-knowledge proofs, a cryptographic method that allows users to prove possession of certain information without revealing it. Zcash used zero knowledge proofs to bring confidential of transactions transactions, while mina used it to have a succinct blockchain it can pass to it's ultra light clients.

In today's lecture, we are going to look at a specific layer 2 scaling solution for ethereum (and all evm compatible chains) called ZK-Rollups.

Introduction:

Ethereum currently handles around 30 transactions per second. Layer 2 scaling solutions for Ethereum are a set of techniques that aim to increase the network's capacity to handle more transactions per second (TPS) and reduce the cost of transactions. These solutions work by creating a new layer on top of the Ethereum network that operates independently and facilitates faster and cheaper transactions.

There are several Layer 2 scaling solutions available for Ethereum, including **state channels**, **sidechains**, **zk-rollups**, **optimistic rollups**, **validium**.

State channels: The equivalent of the lightning netowork on ethereum. A state channel is a technique that enables off-chain transactions between two parties, where they can interact with each other without broadcasting the transaction to the main Ethereum network. Once the transaction is completed, the final state of the transaction is submitted to the Ethereum network. This approach reduces the number of transactions that need to be processed on the main Ethereum network, thereby increasing the network's scalability. This approach doesn't work for general purpose contracts.

Sidechains: A sidechain is a separate blockchain that runs parallel to the Ethereum network and operates with its own set of rules. Transactions on a sidechain do not require the consensus of the main Ethereum network, which allows for faster and cheaper transactions. Once the transaction is completed on the sidechain, the final state is reconciled with the main Ethereum network. (Polygon is a famous ethereum sidechain). This approach does not benefit from the security of ethereum. Instead, it relies on the security of the sidechain.

Rollups: Rollups are a technique that bundles multiple transactions into a single transaction and submits it to the Ethereum network. This approach reduces the number of transactions that need to be processed by the main Ethereum network and significantly reduces transaction fees. There are two types of rollups with different security models:

Optimistic rollups: assumes transactions are valid by default and only runs computation, via a fraud proof, in the event of a challenge.

Zero-knowledge rollups: runs computation off-chain and submits a validity proof to the chain. More on zero-knowledge rollups

Rollups are fully general-purpose, and one can even run an EVM inside a rollup, allowing existing Ethereum applications to migrate to rollups with almost no need to write any new code.

Validium is a scaling solution that enforces integrity of transactions using validity proofs like ZK-rollups, but doesn’t store transaction data on the Ethereum Mainnet. While off-chain data availability introduces trade-offs, it can lead to massive improvements in scalability (validium can process ~9000 transactions, or more, per second)

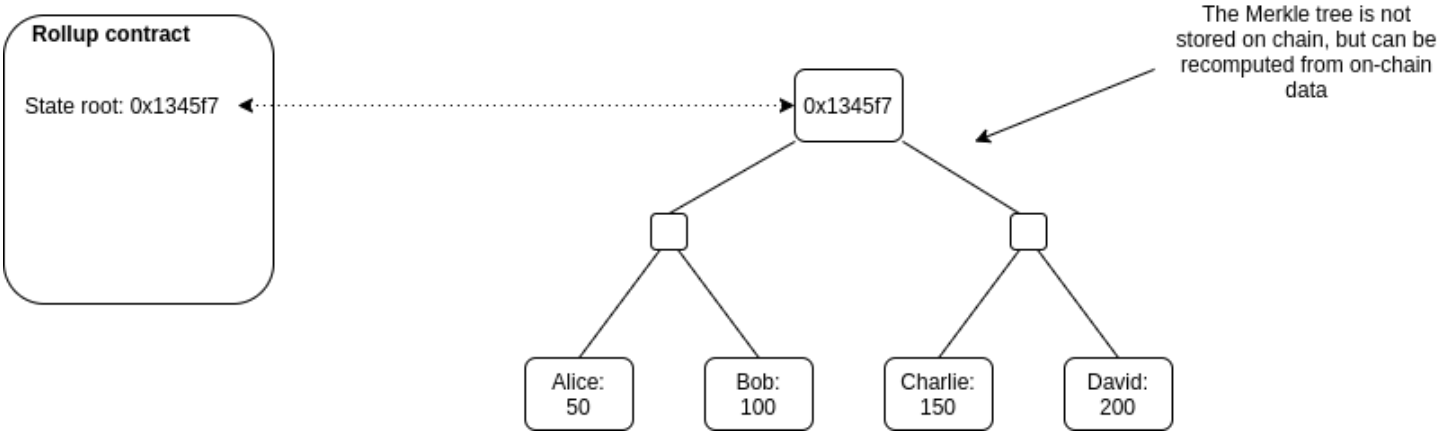
How exactly does a rollup work?

Rollups core architecture is made up of the following components:

On-chain contracts:ZK-rollup protocol is controlled by smart contracts running on Ethereum. This includes the main contract which stores rollup blocks, tracks deposits, and monitors state updates. In case for zk-rollups, another on-chain contract (the verifier contract) verifies zero-knowledge proofs submitted by block producers. Thus, Ethereum serves as the base layer or "layer 1" for the ZK-rollup.

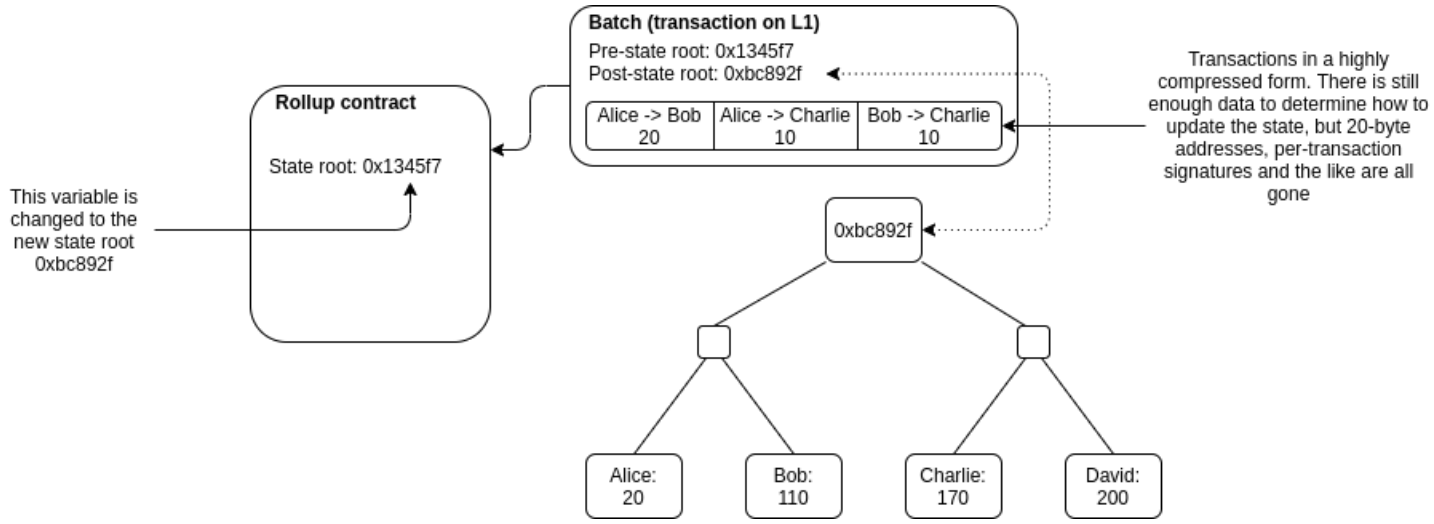
Off-chain virtual machine (VM): While the ZK-rollup protocol lives on Ethereum, transaction execution and state storage happen on a separate virtual machine independent of the EVM. This off-chain VM is the execution environment for transactions on the ZK-rollup and serves as the secondary layer or "layer 2" for the ZK-rollup protocol. In zk-rollups, validity proofs are asubmitted and verified on Ethereum Mainnet to guarantee the correctness of state transitions in the off-chain VM.

There is a smart contract on-chain which maintains a state root: the Merkle root of the state of the rollup (meaning, the account balances, contract code, etc, that are "inside" the rollup).



In order to facilitate the process of depositing and withdrawing into and out of the rollup, if a batch includes external inputs, the transaction that submits the batch must transfer these assets to the rollup contract. Conversely, if a batch includes external outputs, the smart contract will initiate those withdrawals during batch processing.

The operator (usually a centralized party) posts a new state root (together with the old state root) and one new compressed bundle transaction.



How to prove that the Post state root is correct?

Zk-rollups: Provide a zk-snark that proves that all the transactions inside the rollup are valid.

Optimistic rollup: Assumes that the trasactions are valid. If someone detects that there might be fraud, they can post a **fraud proof** to prove that the operator is malicious.

Property	Optimistic rollups	ZK rollups
Fixed gas cost per batch	**~40,000** (a lightweight transaction that mainly just changes the value of the state root)	~500,000 (verification of a ZK-SNARK is quite computationally intensive)
Withdrawal period	~1 week (withdrawals need to be delayed to give time for someone to publish a fraud proof and cancel the withdrawal if it is fraudulent)	**Very fast** (just wait for the next batch)
Complexity of technology	**Low**	High (ZK-SNARKs are very new and mathematically complex technology)
Generalizability	**Easier** (general-purpose EVM rollups are already close to mainnet)	Harder (ZK-SNARK proving general-purpose EVM execution is much harder than proving simple computations, though there are efforts (eg. Cairo) working to improve on this)
Per-transaction on-chain gas costs	Higher	**Lower** (if data in a transaction is only used to verify, and not to cause state changes, then this data can be left out, whereas in an optimistic rollup it would need to be published in case it needs to be checked in a fraud proof)
Off-chain computation costs	**Lower** (though there is more need for many full nodes to redo the computation)	Higher (ZK-SNARK proving especially for general-purpose computation can be expensive, potentially many thousands of times more expensive than running the computation directly)

vitalik's blog

How does compression work?

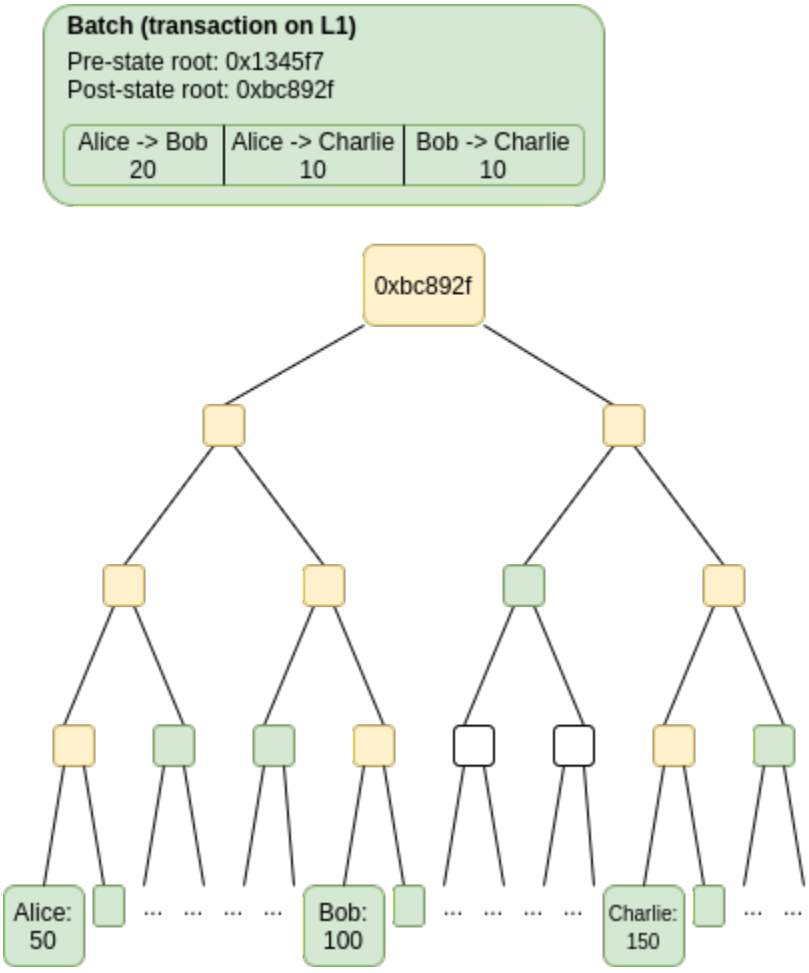
Below is a table comparing how much an ethereum transaction would cost (on average) with and without rollups.

Parameter	Ethereum	Rollup
Nonce	~3	0
Gasprice	~8	0-0.5
Gas	3	0-0.5

Parameter	Ethereum	Rollup
To	21	4
Value	~9	~3
Signature	~68 (2 + 33 + 33)	~0.5
From	0 (recovered from sig)	4
Total	~112	~12

The main savings comes from batchin multiple signatures into one and referring to addresses per index instead of their full address.

How does a fraud proof work (optimistic rollup)?



Assume that an operator or a sequencer published a a Post state root that is fraudulant. Any party that is monitoring the transactions on both L1 and L2 chain can figure out that the state root is computed in a wrong way. The party shows the smart contract on L1 that the Post-State root has been computed in a wrong way by providing the disputed transaction(s). The L1 verifier contract re-executes the transaction(s) (remember that the verifier contract has access to the pre-state root as well) and decides who wins based on that.

In case of fraud, the operator is slashed.

For this reason, optimistic rollups specify a time window where parties can challenge the state of the rollup. As a consequence, withdrawal and deposits into the rollup take a lot of time (a round a week).

Notice that the security of optimistic rollups relies on having **at least one honest party** monitoring the chain. In practice, game theory and incentives play a roll here. The optimistic rollup designers need to give incentive for parties to monitor the chain.

How does a validity Proof work (ZK-rollup)?

Whenever a sequencer or an operator publishes a new post-state root, they also have to submit a zero-knowledge proof that the state has been computed correctly. The sequencer computes the state by running the offchain-evm on the transactions posted by the users. The resulting state root is proven to be correct by using a zk-snark proof that proves that the computation has been correctly. The proof is then verified onchain by the L1 contract.

In comparision with optimistic rollups this offers two benefits:

No waiting time for withdrawals and deposits (just wait for the next batch)

Security against bad sequencers are guaranteed because they can't generate bad proofs.

Censorship resistance.

Many rollups utilize an operator or sequencer to perform tasks such as executing transactions, creating batches, and submitting blocks to L1. While this approach is efficient, it also raises concerns about censorship as dishonest rollup operators may refuse to include certain transactions in batches, thereby censoring users.

To prevent such scenarios, rollups have a security mechanism in place that enables users to directly submit transactions to the rollup contract on Mainnet if they suspect censorship by the operator. This lets users exit the rollup and transfer their assets to Ethereum without requiring the operator's approval.

Pros and Cons of rollups:

Optimistic rollups:

Pros	Cons
Offers massive improvements in scalability without sacrificing security or trustlessness.	Delays in transaction finality due to potential fraud challenges.

Pros	Cons
Transaction data is stored on the layer 1 chain, improving transparency, security, censorship-resistance, and decentralization.	Centralized rollup operators (sequencers) can influence transaction ordering.
Fraud proving guarantees trustless finality and allows honest minorities to secure the chain.	If there are no honest nodes a malicious operator can steal funds by posting invalid blocks and state commitments.
Computing fraud proofs is open to regular L2 node, unlike validity proofs (used in ZK-rollups) that require special hardware.	Security model relies on at least one honest node executing rollup transactions and submitting fraud proofs to challenge invalid state transitions.
Rollups benefit from "trustless liveness" (anyone can force the chain to advance by executing transactions and posting assertions)	Users must wait for the one-week challenge period to expire before withdrawing funds back to Ethereum.
Optimistic rollups rely on well-designed cryptoeconomic incentives to increase security on the chain.	Rollups must post all transaction data on-chain, which can increase costs.
Compatibility with EVM and Solidity allows developers to port Ethereum-native smart contracts to rollups or use existing tooling to create new dapps.	

Source: vitalik's blog

Zk-rollups:

Pros	Cons
Validity proofs ensure correctness of off-chain transactions and prevent operators from executing invalid state transitions.	The cost associated with computing and verifying validity proofs is substantial and can increase fees for rollup users.
Offers faster transaction finality as state updates are approved once validity proofs are verified on L1.	Building EVM-compatible ZK-rollups is difficult due to complexity of zero-knowledge technology.
Relies on trustless cryptographic mechanisms for security, not the honesty of incentivized actors as with optimistic rollups.	Producing validity proofs requires specialized hardware, which may encourage centralized control of the chain by a few parties.
Stores data needed to recover the off-chain state on L1, which guarantees security, censorship-resistance, and decentralization.	Centralized operators (sequencers) can influence the ordering of transactions.
Users benefit from greater capital efficiency and can withdraw funds from L2 without delays.	Hardware requirements may reduce the number of participants that can force the chain to make progress, increasing the risk of malicious operators freezing the

Pros	Cons
	rollup's state and censoring users.
Doesn't depend on liveness assumptions and users don't have to validate the chain to protect their funds.	Some proving systems (e.g., ZK-SNARK) require a trusted setup which, if mishandled, could potentially compromise a ZK-rollup's security model.
Better data compression can help reduce the costs of publishing `calldata` on Ethereum and minimize rollup fees for users.	

Source: vitalik's blog