Rollups

A learning group for ZK and SNARK application development

Daniel Szego

In: https://www.linkedin.com/in/daniel-szego/

Logistics: ZK Learning Group

Every month, third thursday in 2025, from 18 (CET)

One hour, presentation + short discussion

Different topics on zero knowledge proof,

- mostly from programmer and application developers perspective
- with some theory

Coordination:

- Discord channel: LF Decentralized Trust
  https://discord.com/channels/905194001349627914/1329201532628898036
- Meetup.com: https://www.meetup.com/lfdt-hungary/events/305634614/
- Repo with all the contents:https://github.com/LF-Decentralized-Trust-labs/
  https://github.com/LF-Decentralized-Trust-labs/zk-learning-group

Quizzes and small programming challenges, LFDT merchs at the end

Logistics: Hunting for the SNARK

February - Introduction, Theory : Definitions and building blocks

March - Theory : Polynomial commitments

April - Theory : Interactive oracle proofs

May - Programming : Circom

June - Programming : Circom

July - Programming : Noir - basics

August - Programming : Noir - advanced

September : Applications : Proof of reserve, proof of solvency

October : Applications : ZK machine learning

November : Applications : Rollup

December : Wrap up, Applications

Subject to change based on community discussion ….

Agenda

- *Zero knowledge proofs*
- *Blockchain scaling*
- *Optimistic Rollup*
- *ZK Rollup*
- *Optimistic vs ZK Rollup*
- *Zk proof for off-chain execution*
- *Rollup: off-chain transactions*
- *ZK Rollup high level overview*
- *Naysayer proof and rollups*
- *Challenge*
- *Links and resources*
- *Q&A*

Zero knowledge proofs

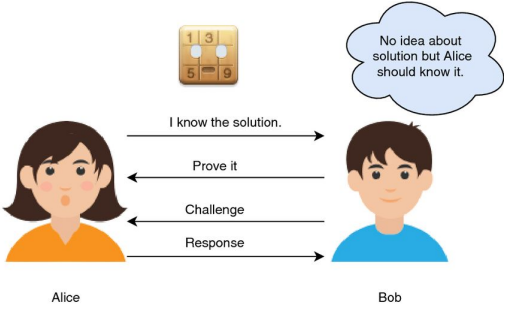
Computation: arithmetic circuit : *C(x, w) -> F*
- *x* public input
- *w* private input, witness
- high level computation
- arithmetic circuit
- polynomials

Prover algorithm: *P (pp, x, w) -> proof*

Verifier algorithm: *V (vp, x, proof) -> accept / reject*

Properties:

- *Succinct:*
- *Complete:*
- *Knowledge sound:*
- *Zero knowledge*



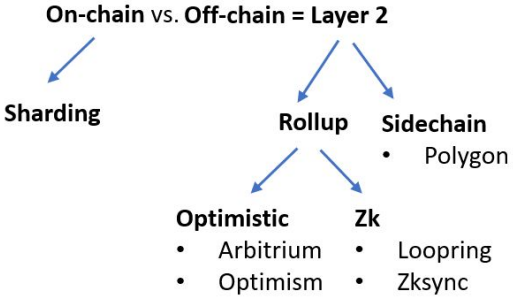
Blockchain scaling

Scalability trilemma:

- Security
- Decentralization
- Scalability

Scaling direction:

- On-chain scaling
- Different consensus algorithms
- Sharding
- Sidechains, parachains
- Off-chain (L2) scaling
- State channels, lightning network
- Rollups



Optimistic Rollup

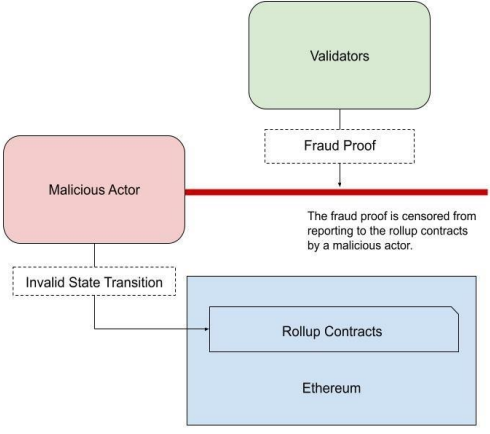
Transactions are executed **off-chain** (rollup / sequencer)

Transaction **result** is **imported back** into the blockchain

Optimistic assumption: the result is correct unless:

- **Challenge period** (e.g. 1 week)
- **Fault proof** by independent validators
- Fault proof evaluation is on the blockchain (smart contract)
- State is finalized after the challenge period (no fault proof)
- **Incentivization**: stake, slashing

E.g. Optimism



ZK Rollup

Transactions are executed off-chain

Rollup, sequencer

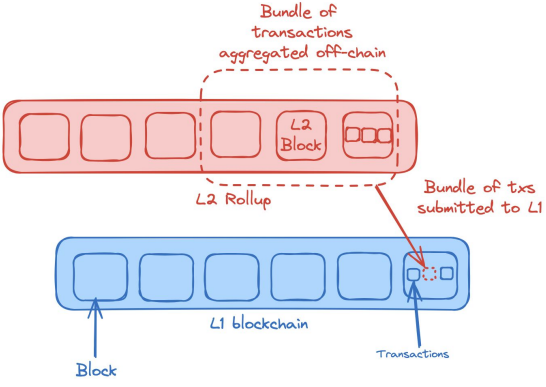
Execution result is exported back to the blockchain

Execution result is hardened by a zero knowledge proof

Zero knowledge proof if validated on-chain (smart contract)

Result is valid only if the ZK proof on the correct execution is valid as well.

E.g. Arbitrum



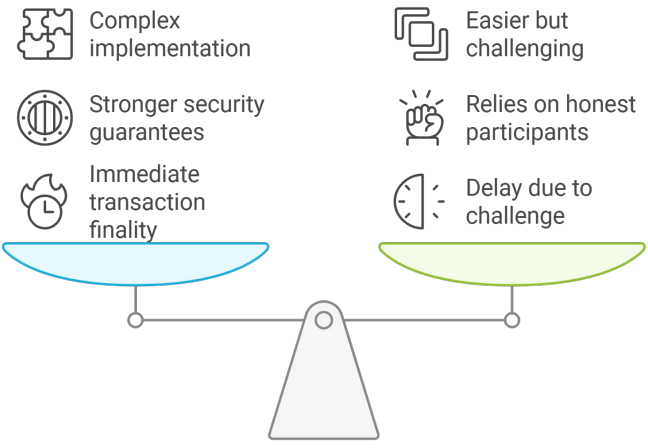
Optimistic vs ZK rollup

Optimistic:

- economical security guarantee
- long finality time
- honest challenger

ZK:

- cryptographic security guarantee
- fast finality
- performance issues (prove / verify)

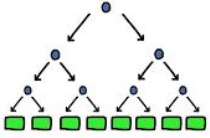


ZK Proof for off-chain execution

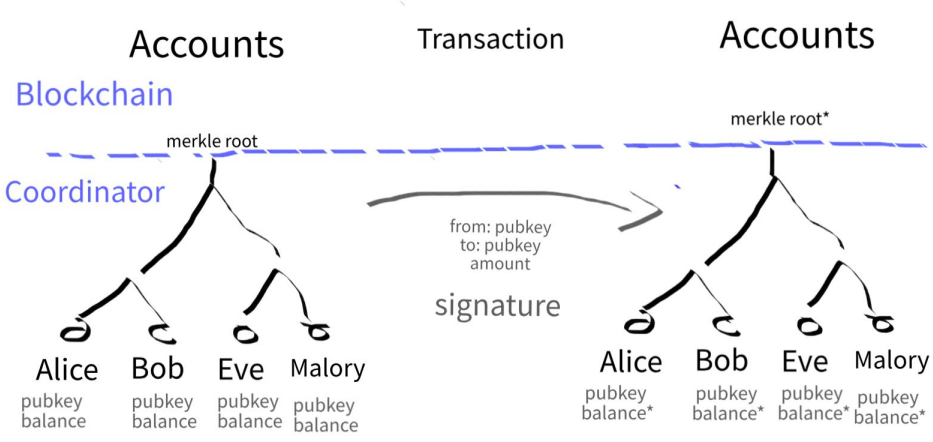
Accounts are represented as leafs in the merkle tree.

Transaction validation steps (simplified):

- Transaction : send *<from>* , *<to>*, *<amount>*
- check if *<from>* and *<to>* are on the account tree, which merkle root is already in the blockchain
- check if *<from>* has enough balance
- debit *<from>* with *<amount>*
- credit *<to>* with the *<amount>*
- calculate new account tree and merkle root for it
- create zk proof about the calculation publish new root + zkProof to the ledger
- if the proof is valid, record new root to in the ledger



Rollup: off-chain transactions



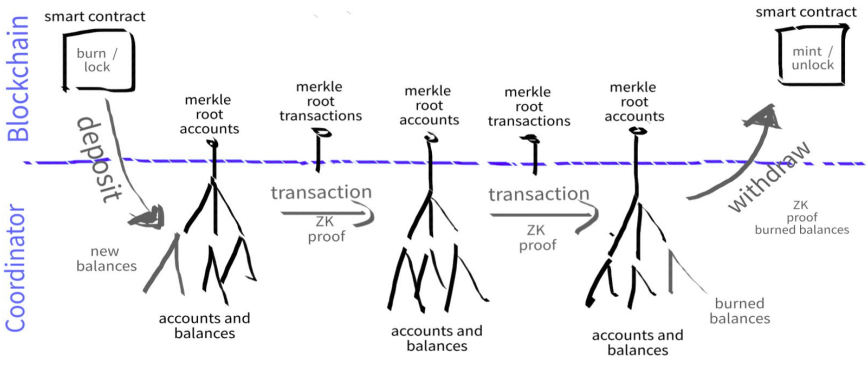
ZK Rollup high level overview

Off-chain transactions

Payment / token transfer

Transactions are executed in batch / block by a coordinator

Deposit and withdraw transactions



ZK rollup blog: https://github.com/tanpx12/zk-rollup-tutorial

Naysayer proof and rollups

Hybrid approach

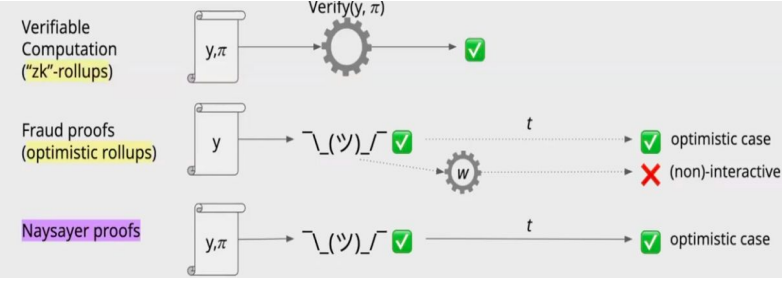
Optimistic and ZK rollup combination

ZK proof is published on-chain, but not verified

Published ZK proof can be challenged in an optimistic style with fraud proofs

Systems with very limited on-chain computational resources

E.g. BitVM



Challenge

Create a simple ZK rollup in circom or in noir

Links, Resources, Literature

zk-rollup-tutorial
https://github.com/tanpx12/zk-rollup-tutorial

Build Your Own Rollup
https://medium.com/l2beat/build-your-own-rollup-72423f4255e7

Optimistic Rollups
https://ethereum.org/developers/docs/scaling/optimistic-rollups/

Zero-knowledge rollups
https://ethereum.org/developers/docs/scaling/zk-rollups/

How to Implement a Minimalist NFT zkRollup With Circom and SnarkJS
https://hackernoon.com/how-to-implement-a-minimalist-nft-zkrollup-with-circom-and-snarkjs

Happy Hunting for the SNARK :)

Q & A

Daniel Szego

In: https://www.linkedin.com/in/daniel-szego/