Proof of reserve
Proof of solvency

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/Daniel-Szego/zk-leraning-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 : Off-chain transactions

November : Applications : Rollup

December : Wrap up, Applications

Subject to change based on community discussion ….

Agenda

- *Zero knowledge proofs*
- *Centralized exchanges, centralized custodians*
- *Proof of reserve*
- *Proof of liability*
- *Proof of solvency*
- *Attack model*
- *Demo*
- *Challenge*
- *Links, Resources, Literature*
- *Q&A*

Zero knowledge proofs

"Proof" of a statement, e.g. I know a preimage of a hash function

It's not a "classic" mathematical proof, it's s know with high probability

I know some kind of secret information, I "p know without saying it

Roles:

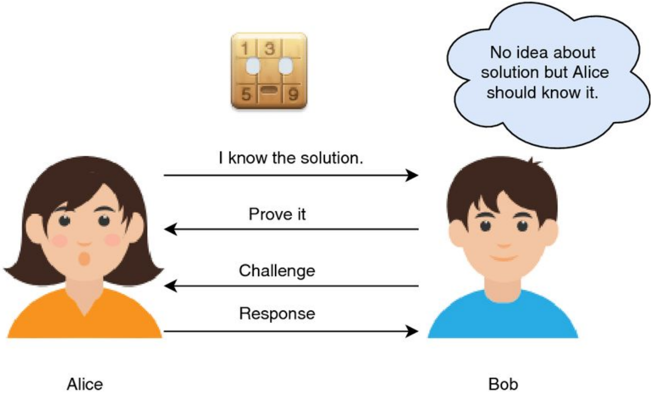
- Prover: prover

- Verifier: verifier, validator

Interactive / non-interactive

Proof size, prover / verifier time

(zk)SNARK: Succinct Non-interactive ARgument of Knowledge



Centralized exchanges, centralized custodians

Centralized crypto custodians

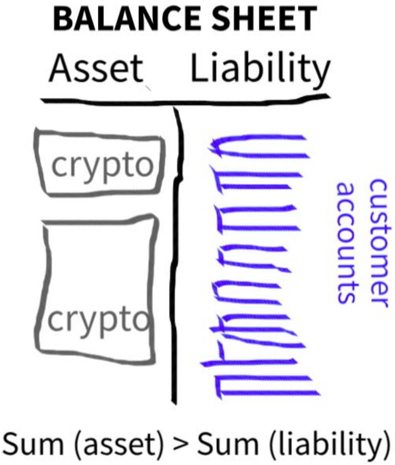
Asset:

- Store crypto as reserve
- Different cryptocurrency
- Different cryptocurrency wallets

Liability:

- Web2 user experience for customers
- Customer accounts
- Custodial (Web2) wallets
- Customer balances == “liabilities”

Sum of all assets should be bigger than all liabilities.



Proof of reserve

Cryptocurrency addresses : publicly check balances

Reveal addresses: critical information leak

Proving the amount of reserve without revealing the addresses of the custodian

Zero knowledge proof

Account balance bases systems

Ownership of and address : private key ownership

Account and balance is stored in a Merkle Patricia tree of the given blockchain

Proof of liability

Customer accounts - balances are not on the blockchain

Custodian owns with the balances

Customer identification and balance

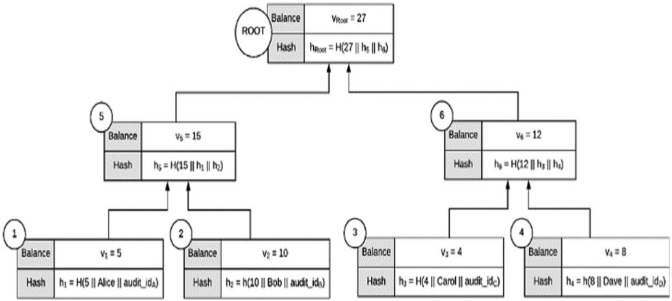
Merkle tree

Summation merkle tree with aggregated balances

Proof if my account and balances is contained in the tree

Sum of all the accounts, equals all the liabilities

Proof by summation merkle tree


ROOT
Balance
$V_{Root} = 27$
Hash
$h_{Root} = H(27 || h_5 || h_6)$
5
Balance
$V_5 = 15$
Hash
$h_5 = H(15 || h_1 || h_2)$
6
Balance
$V_6 = 12$
Hash
$h_6 = H(12 || h_3 || h_4)$
1
Balance
$V_1 = 5$
Hash
$h_1 = H(5 || Alice || audit\_id_A)$
2
Balance
$V_2 = 10$
Hash
$h_2 = h(10 || Bob || audit\_id_B)$
3
Balance
$V_3 = 4$
Hash
$h_3 = H(4 || Carol || audit\_id_C)$
4
Balance
$V_4 = 8$
Hash
$h_4 = h(8 || Dave || audit\_id_D)$

Proof of solvency

Combining proof of liability with proof of reserve

Proving that the reserve is bigger than the liability.

Zero knowledge proof

Hiding the exact amount of reserve and liability

Hiding the exact crypto addresses

Customer addresses and balances are visible only to the given customers

Proof might be “big” - no on-chain verification : e.g. STARK

Attack models

Querying the liability depends on the customer

if it is not queried, it might be omitted from the liability tree

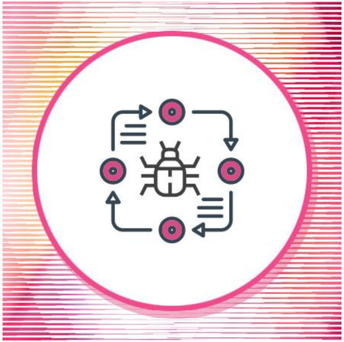
Negative amount and balance might be added to decrease the total liability -> improved proof of liability with ZKP: all balances are positive

Creating a liability proof is resource intensive :

- if it is done by the provider it can be "gamed"
- If it is done by a third party company

Proof of reserve prove the control of the certain crypto asset, but:

- It might not owned by the company
- It might be used in other scenarios (like lending, double reserve, etc …)



Challenge

Investigate and experiment with a proof of solvency system of an exchange or centralized crypto custodian on your own

Links, Resources, Literature

Design and implementation of solvency proof system based on zero knowledge proofs
https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/blc2.12089

Distributed Auditing Proofs of Liabilities
https://eprint.iacr.org/2020/468.pdf

Proof of Reserves
A Report on Mitigating Crypto Custody Risk
https://www.btcpolicy.org/articles/proof-of-reserves-a-report-on-mitigating-crypto-custody-risk

Private Proof of Solvency
https://arxiv.org/abs/2310.13900

Exploring Proof of Solvency and Liability Verification Systems
https://blog.chain.link/proof-of-solvency/

Happy Hunting for the SNARK :)

Q & A

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