Decentralized AI with ZK machine learning

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

- *zk(SNARK)*
- *Applications*
- *zkML - zero knowledge machine learning*
- *zkML - Private Input, Public Model*
- *zkML - Private Model, Public Input*
- *zkML - Other scenarios*
- *L1 integration*
- *L2, rollup integration*
- *Challenge*
- *Links, Resources, Literature*
- *Q&A*

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

Computation: arithmetic circuit : *C(x, w) -> F*

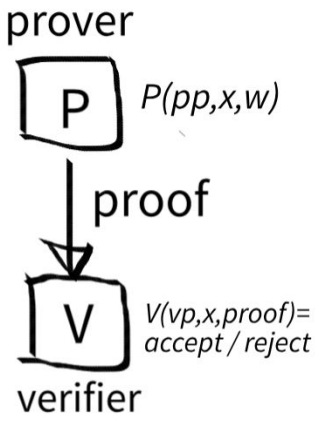
- *x* public input
- *w* private input, witness
- high level computation
- arithmetic circuit
- polynomials

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

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

Properties:

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



zkML - zero knowledge machine learning

Machine Learning:

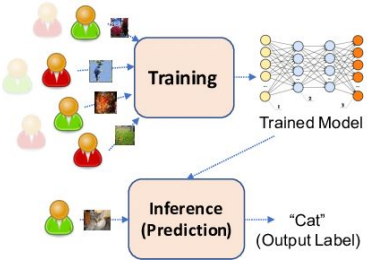
- learning phase:
- supervised / non-supervised learning
- inference: input out association based on the trained model

ML + zkSNARK:

- Trained machine learning model (like off-chain)
- Inference phase is supported by zero knowledge proofs
- Proofs and the results can be validated (like(on-chain)

zkML models (mostly inference):

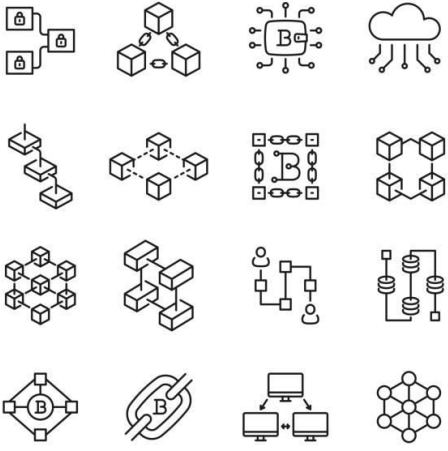
- public / private : trained model / input / output



Applications

Example applications:

- off-chain credit rating results verifiable on-chain
- on-chain asset management: portfolio rebalancing based on verifiable machine learning
- AI driven AML for DeFI protocols, integrating off-chain AI execution results in a verifiable way
- biometric identification for on-chain protocols, hiding relevant biometric information
- processed data integration of IoT devices
- Agentic AI verifiable models



zkML - Private Input, Public Model

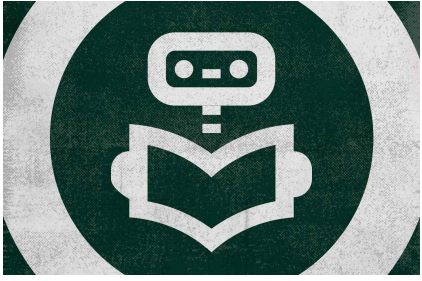
Inference phase is supported by zero knowledge proofs

Public parameters of the trained model

Private input for the on-chain output, but there must be a proof that the output is based on the private input and trained model:

0. commit parameters of the machine learning model publicly.
1. commit private input hash, *hash(x)*
2. give a snark proof that *hash(x)* was committed into the ledger and the output is if we apply the machine learning to x.

E.g. credit scoring



Link: https://0xparc.org/blog/zk-mnist

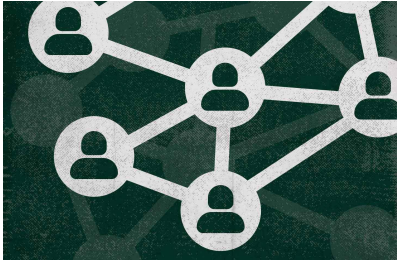
zkML - Private Model, Public Input

The trained machine learning must be kept secret.

It is industry standard or under Intellectual Property (IP) rights

It must be proven that the same model used for different computations:

0. Commit hash of the parameters of the private machine learning model, like *hash(params)*
1. Give a *zk*SNARK proof that the *x* input produces a certain o output by applying the machine learning model which parameters were committed into as *hash(params)*



Link: https://0xparc.org/blog/zk-mnist

zkML - Other scenarios

Private input, private model:

- The input is confidential and model is private as well (like protected by IP).
- E.g. healthcare applications
- Complex: Compositional ZK or multiparty computation

Public input, public model:

- Computational intensive models
- Succinct proof on the computation

Proof of training

Link:
https://medium.com/1kxnetwork/zkml-evolving-the-intelligence-of-smart-contracts-through-zero-knowledge-cryptography-e6725412bbd1

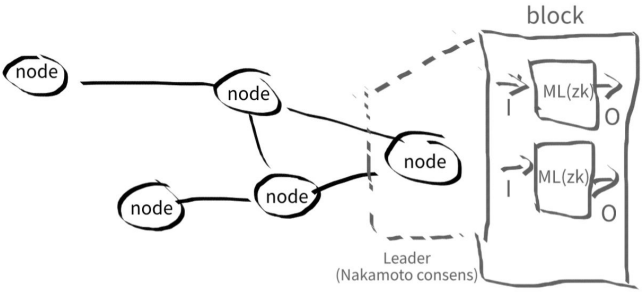
L1 integration

On-chain integration:

- machine learning is realized as smart contracts (scalability problems)

zkML:

- no classical transaction
- deploy machine learning models
- execute machine learning on input, producing output proved by ZK
- optional, learning
- validators validate the correctness of proof
- consensus models: Nakamoto / quorum
- block time considerations ?

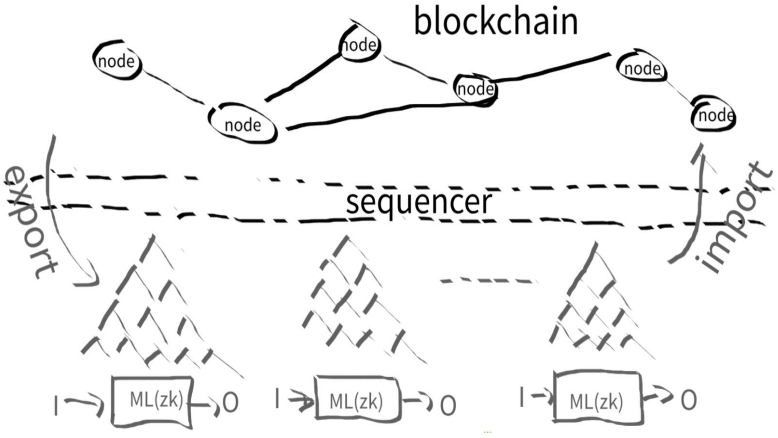


L2, rollup integration

Machine learning is integrated with a L2 subchain

Machine learning is integrated with a L2 rollup:

- deploy ML to rollup
- export data to rollup
- execute inference on the rollup and include ZK proof on.chain
- optional learning
- import the result on-chain



Demo

zkML Demo:

https://zkmnist.netlify.app/

Challenge

Investigate and experiment with a machine learning hardened by zero knowledge proof on Noir or circom

Links, Resources, Literature

ZK Machine Learning
https://0xparc.org/blog/zk-mnist

zkML: Evolving the Intelligence of Smart Contracts Through Zero-Knowledge Cryptography
https://medium.com/1kxnetwork/zkml-evolving-the-intelligence-of-smart-contracts-through-zero-knowledge-cryptography-e6725412bbd1

ZKML: Verifiable Machine Learning using Zero-Knowledge Proof
https://kudelskisecurity.com/modern-ciso-blog/zkml-verifiable-machine-learning-using-zero-knowledge-proof

Research on Zero knowledge with machine learning
https://www.researchgate.net/publication/381544858_Research_on_Zero_knowledge_with_machine_learning

Awesome zkML repository
https://github.com/worldcoin/awesome-zkml

Happy Hunting for the SNARK :)

Q & A

Daniel Szego

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