Lesson 3

Lesson 1 : Fundamentals

Lesson 2 : Introduction to ML

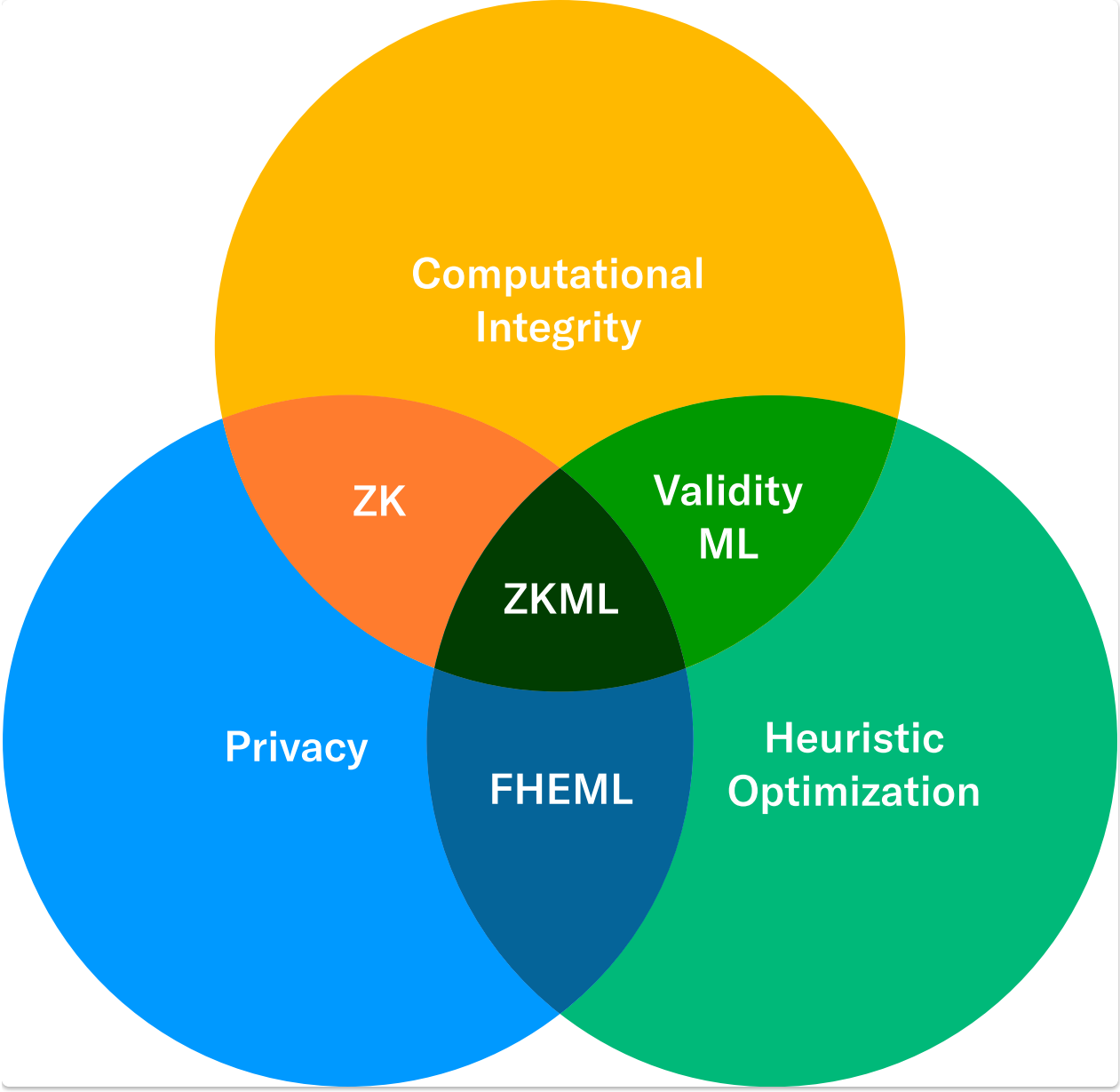
Lesson 3 :Intro to zkML / Use cases

Lesson 4 :EZKL

Today's topics

- zkML introduction
- Ecosystem
- Use cases

> #### zkML Introduction
>
> From introduction blog

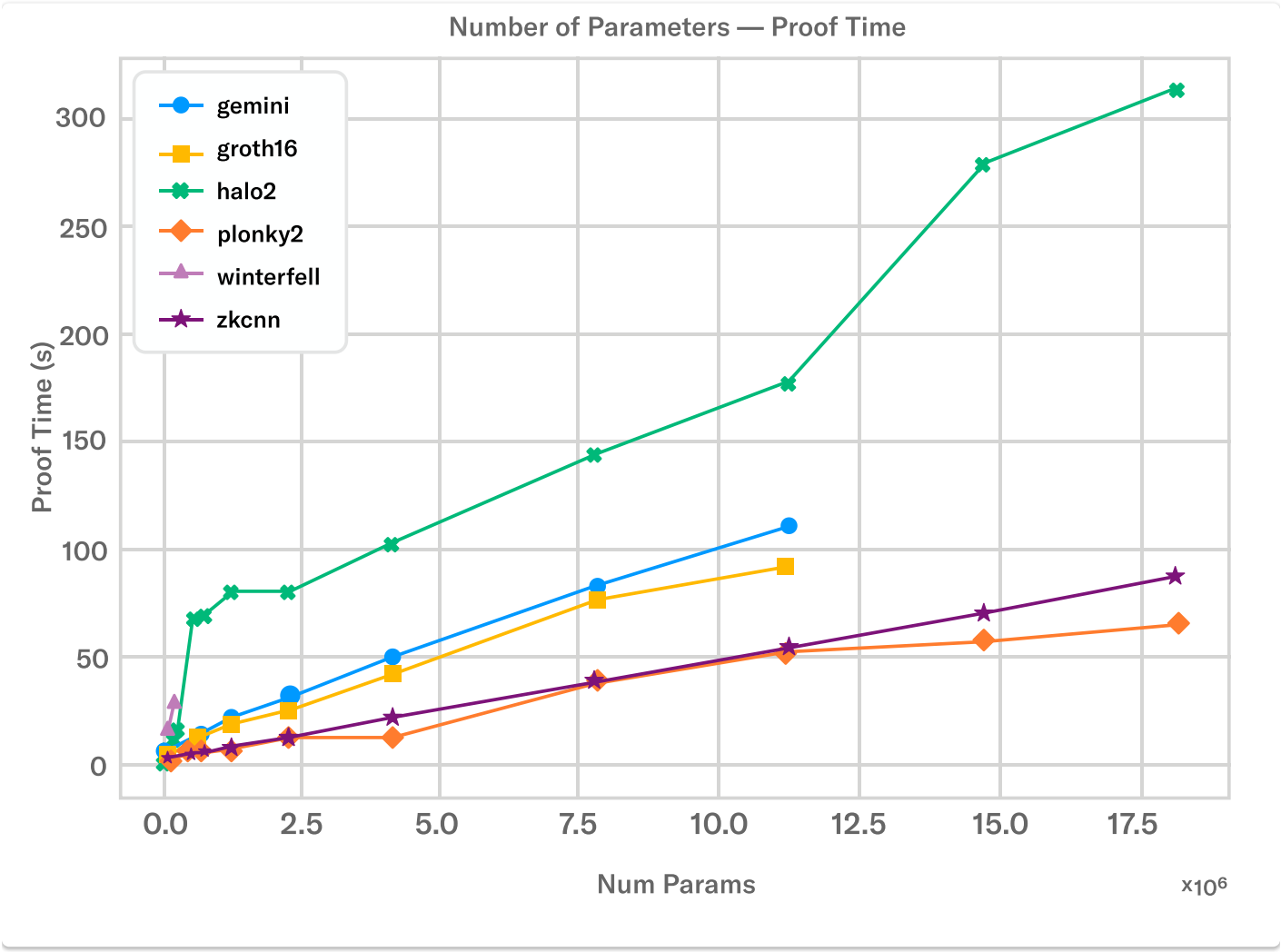


The current state of zero-knowledge systems, even with high-performance hardware, is insufficient to handle large language models. But progress is being made with smaller models.

The Modulus Labs team recently released a paper titled "The Cost of Intelligence", where they benchmark existing ZK proof systems against a wide range of models of different sizes. It is currently possible to create proofs for

models of around 18M parameters in about 50 seconds running on a powerful AWS machine using a proving system like plonky2.

The following illustrates the scaling behaviour of different proving systems as the number of parameters of a neural network are increased:"



Modulus Labs

Modulus Labs are working on a number of use cases :

- On-chain verifiable ML trading bot - RockyBot

- Blockchains that self-improve vision
- Enhancing the Lyra finance options protocol AMM with intelligent features
- Creating a transparent AI-based reputation system for Astraly (ZK oracle)

Federated Machine Learning

See paper

Similar to zkML in its desire for privacy, federated machine learning uses different techniques.
A typical approach would be that a number of users have data on which they would like to collaboratively train a model while still keeping their data private.
To achieve this secure multi party computation or FHE can be used.

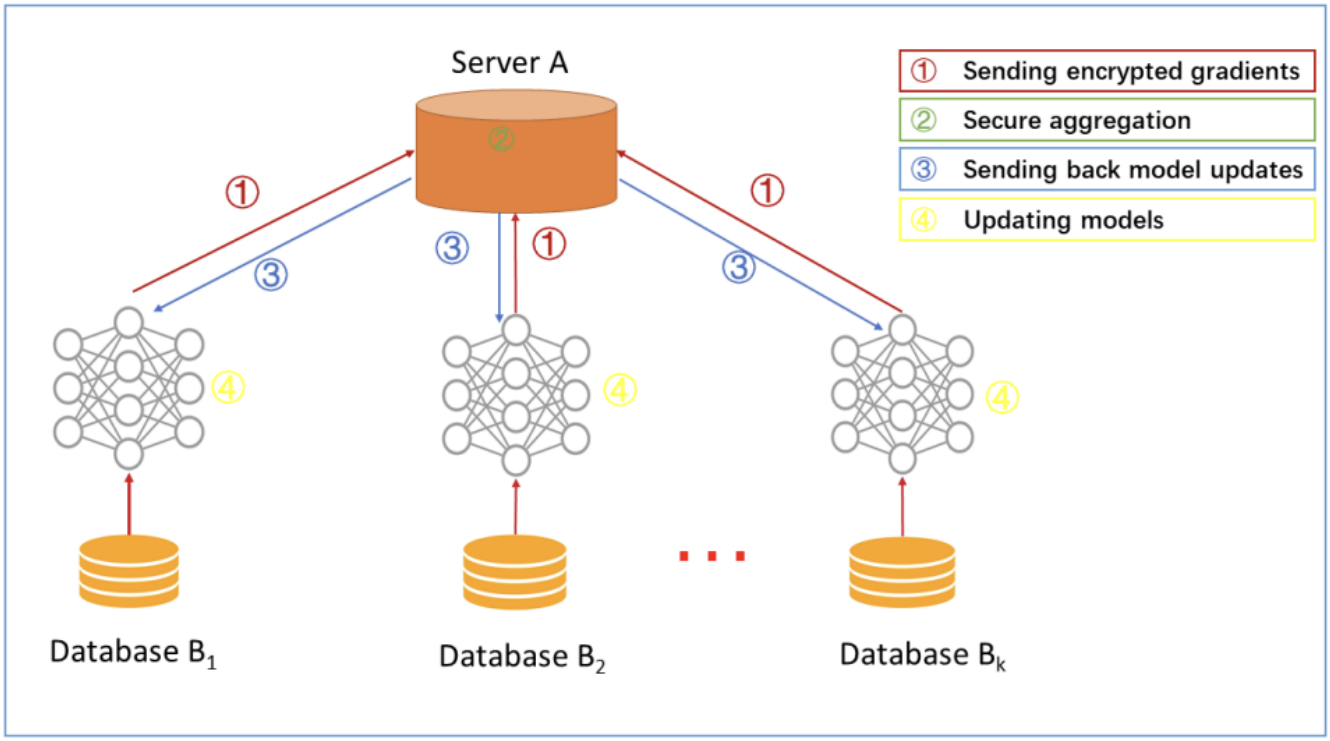
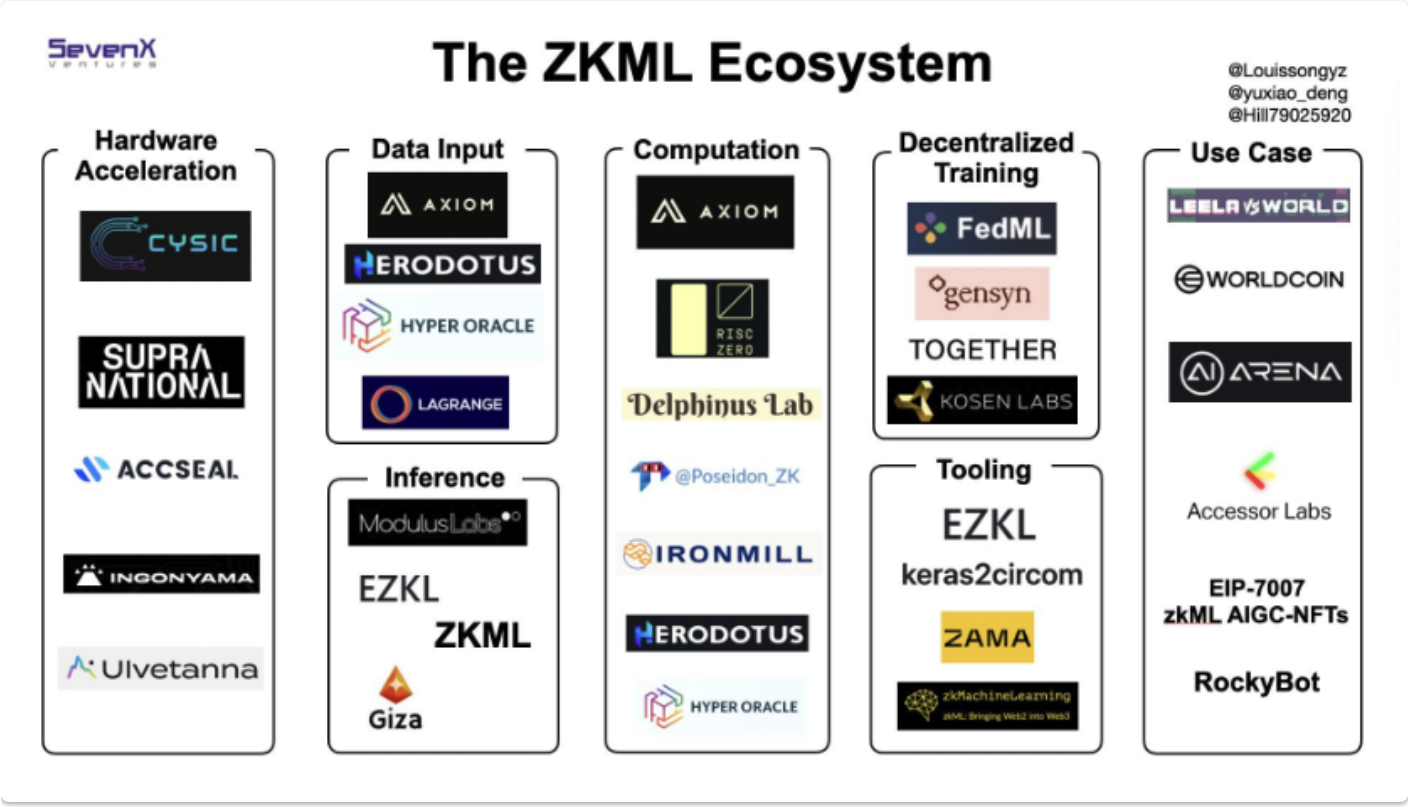


Fig. 3. Architecture for a horizontal federated learning system

Ecosystem


SevenX
The ZKML Ecosystem
@Louissongyz
@yuxiao_deng
@Hill79025920
Hardware Acceleration
CYSIC
SUPRA NATIONAL
ACCSEAL
INGONYAMA
Ulvetanna
Data Input
AXIOM
HERODOTUS
HYPER ORACLE
LAGRANGE
Inference
ModulusLabs
EZKL
ZKML
Giza
Computation
AXIOM
RISC ZERO
Delphinus Lab
@Poseidon_ZK
IRONMILL
HERODOTUS
HYPER ORACLE
Decentralized Training
FedML
gensyn
TOGETHER
KOSEN LABS
Tooling
EZKL
keras2circom
ZAMA
zkMachineLearning
Use Case
LEELA vs WORLD
WORLDCOIN
AI ARENA
Accessor Labs
EIP-7007
zkML AIGC-NFTs
RockyBot

Approaches to modelling a neural network

From article by Pratyush Ranjan Tiwari

1. Naive implementation using addition/multiplication gates: The circuit would be very simple and any proof protocol could be utilised but the size of the circuit is very large $O(n^2 \cdot w^2)$ for image matrix being $n \times n$ and the weight matrix being $w \times w$. Making this method inefficient for any powerful CNN with more than a couple layers.
2. Compute convolutions using FFT: This was demonstrated by an S&P (Oakland) '20 paper ZXZS20 and results in a circuit of size $O(n^2 \log n)$ with $O(\log n)$ depth. Another variant of this approach was followed by the zkCNN paper by utilizing a sumcheck protocol for FFT, which is asymptotically optimal with prover time $O(n^2)$.
3. Convolution $\approx$ Polynomial multiplications: This approach was demonstrated by the vCNN paper. It uses the elegant approach

of first computing the result of the convolution outside the circuit and then given the result as input, it checks equality of polynomial multiplication.
The equality check is done by checking equality and random points, the security comes from the Schwartz-Zippel Lemma
This is the most efficient approach as the zk proof circuit evaluating polynomials at random points is of size $O(n^2 + w^2)$.

A major problem with these approaches is the need to convert floating point numbers to integers for the proofs.
Similarly efficiently proving the correctness of matrix multiplication is difficult.

Some possible workarounds are 'quantise' the weights so that they can be represented as elements in a finite field. This still has the problem of the modular nature of the field.

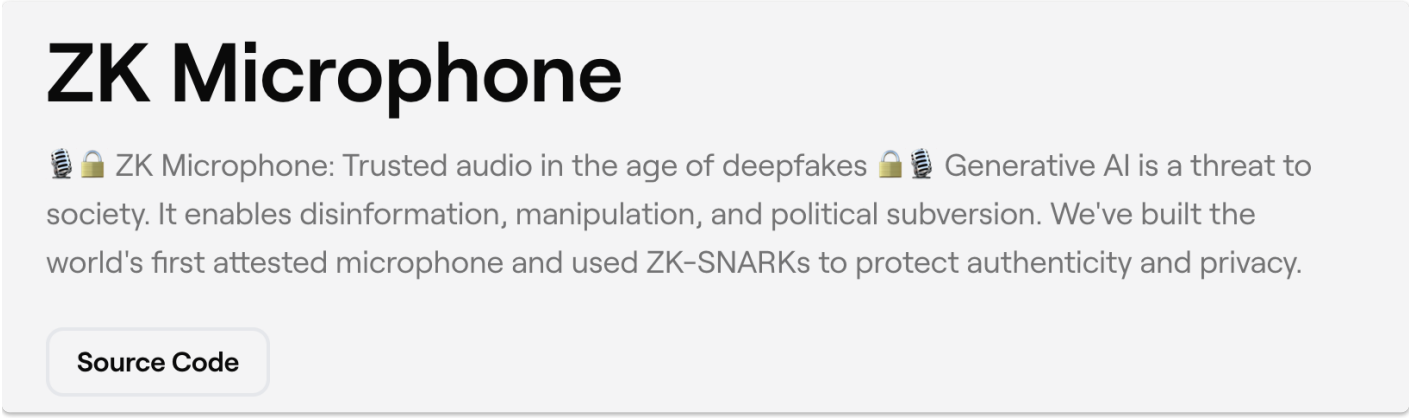
Applications

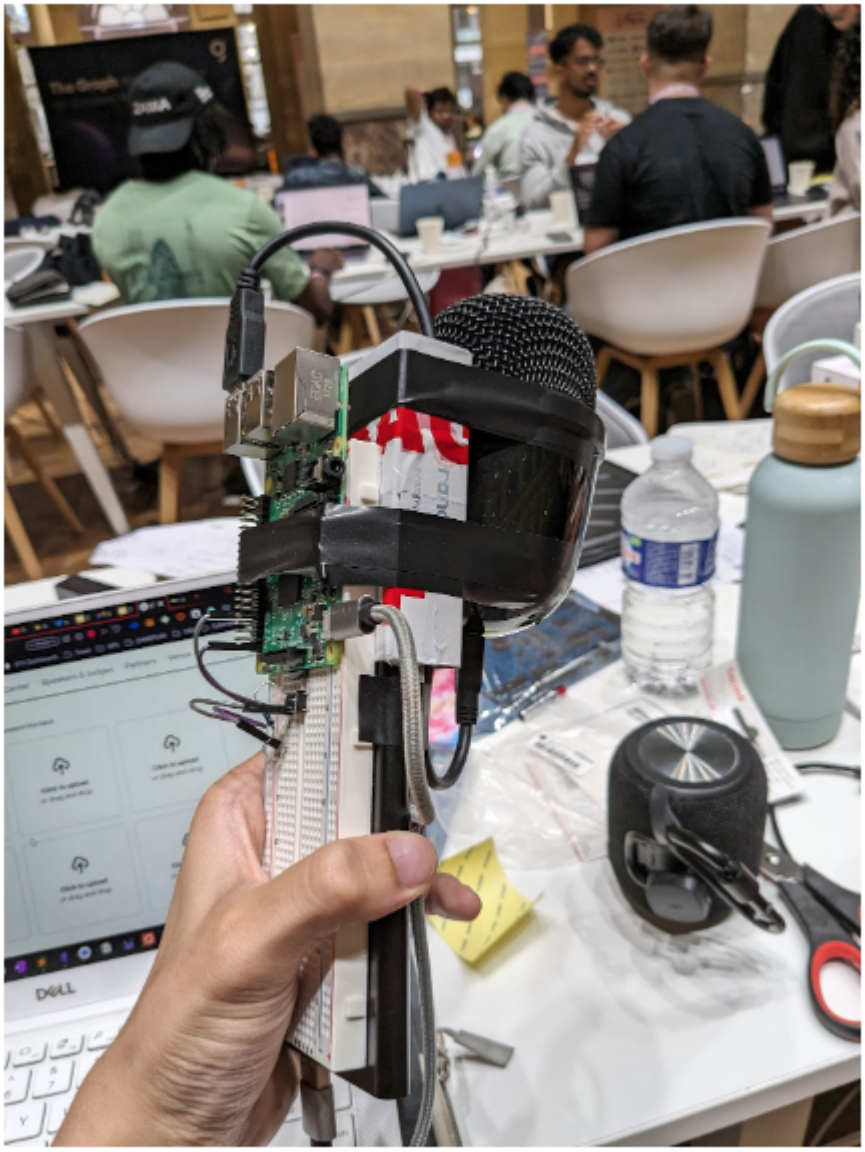
Applications of zkML will typically make use of the features of ZKPs, namely

- proof of correctness of computation
- verifiable computation with private inputs

Media authenticity

Daniel Kang is working on verifying image authenticity.
A similar project is the ZK Microphone.
See recent project from ETH Global



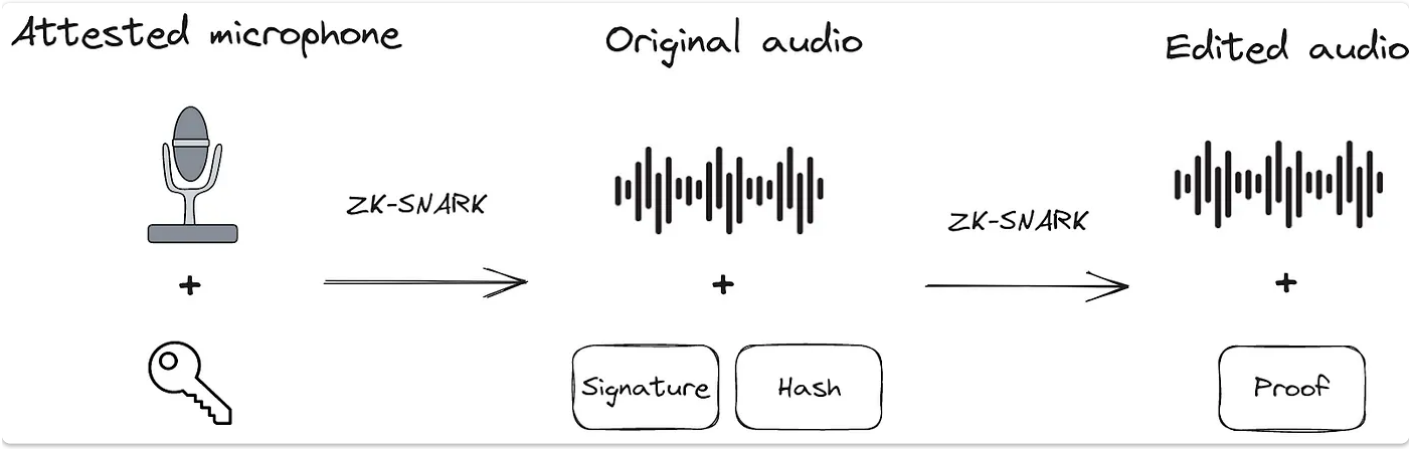


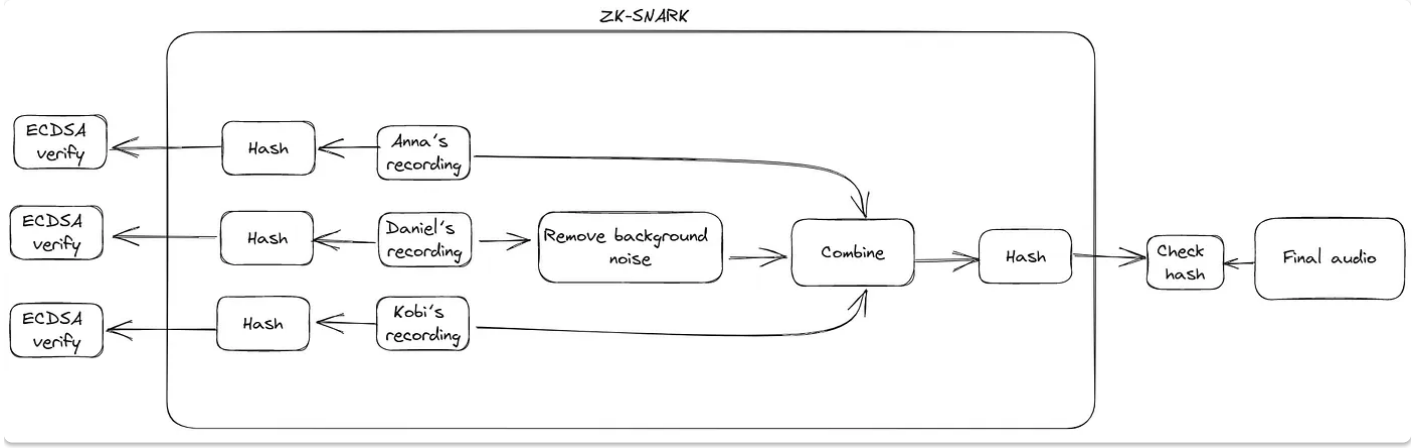
Also see this talk

Attested Audio

See Blog
and Podcast

This was based on a project that generated answers to questions about zk using the voices of the presenters from the zk podcast.
The attested audio tried to show that a clip of audio recorded from the real presenters wasn't a generated clip.
In this example the real audio was edited to remove background noise, but SNARKs used to prove that this was being done correctly and was based on original audio.




ZK-SNARK
ECDSA verify
Hash
Anna's recording
ECDSA verify
Hash
Daniel's recording
Remove background noise
Combine
Hash
Check hash
Final audio
ECDSA verify
Hash
Kobi's recording

ZKaggle

See repo
"Our browser-based frontend allows bounty providers to upload their data to Filecoin and set up a computing task with bounty rewards. Bounty hunters can browse all open bounties, download the data onto their local machine, and compute.
When they are ready to submit, they construct a ZK proof to submit the hashed computed results on-chain."

Nuclear Treaty Verification

See details

An old project, but it shows how ML and ZKP can be used together

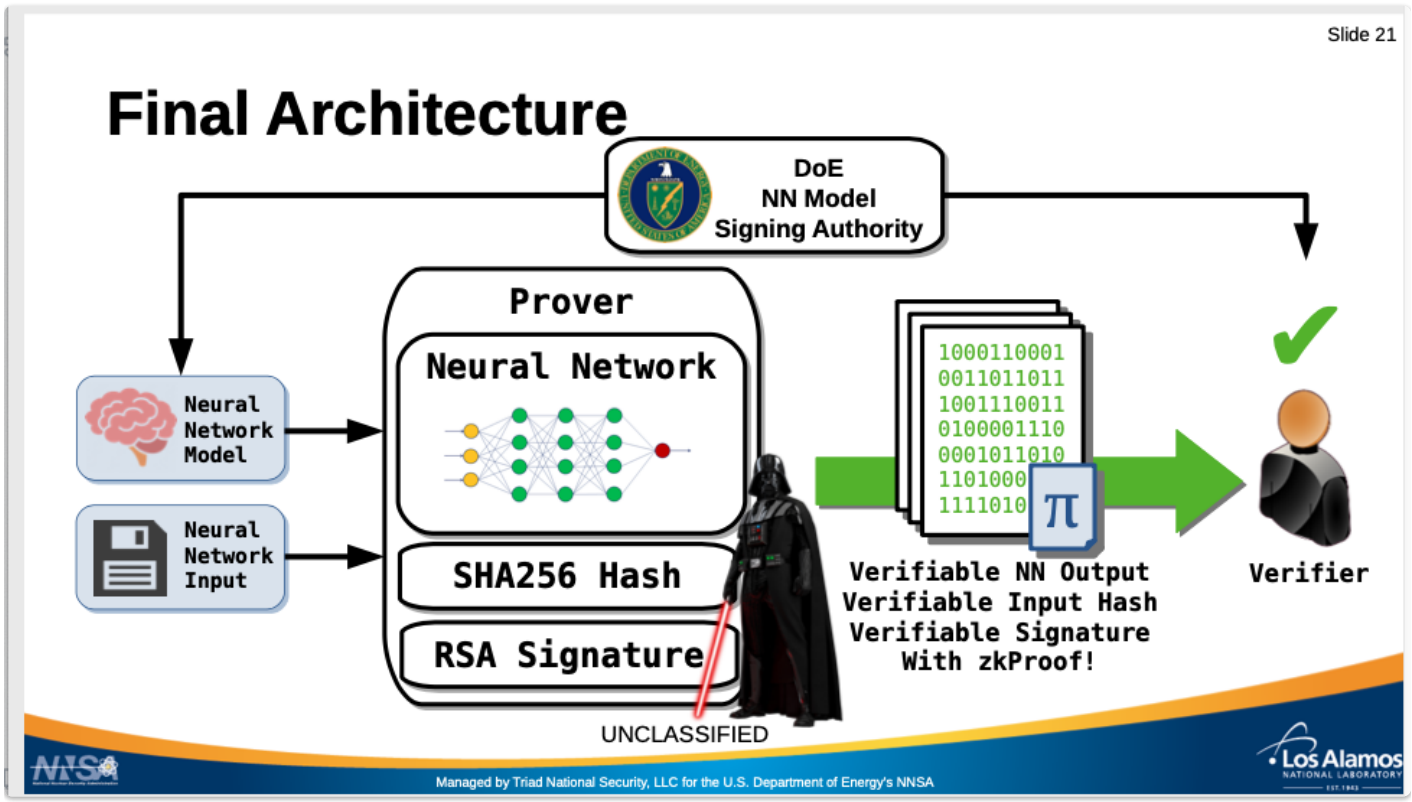
Los Alamos NATIONAL LABORATORY EST.1943

LA-UR-20-20260

Approved for public release; distribution is unlimited.

Title: SNNzkSNARK An Efficient Design and Implementation of a Secure Neural Network Verification System Using zkSNARKs

Author(s): DeStefano, Zachary Louis



Gensys

Decentalised computation