Basics of Noir programming

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 - advance

September : Applications : Off-chain transaction

October : Applications : Proving solvency

November : Applications : Rollup

December : Wrap up, Applications

Subject to change based on community discussion ….

Agenda

- *zkSNARK*
- *ZK languages and tools*
- *Noir concepts*
- *ACIR*
- *Noir programming*
- *Aztec*
- *Aztec demo*
- *Challenge*
- *Lins and resources*
- *Q&A*

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

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

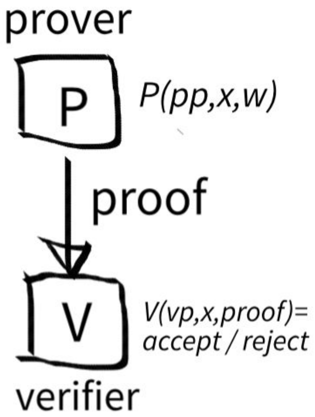
- *x* public input
- *w* private input, witness
- high level computation
- arithmetic circuit
- polynomials

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

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

Properties:

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



ZK languages and tools

Core algorithm consideration

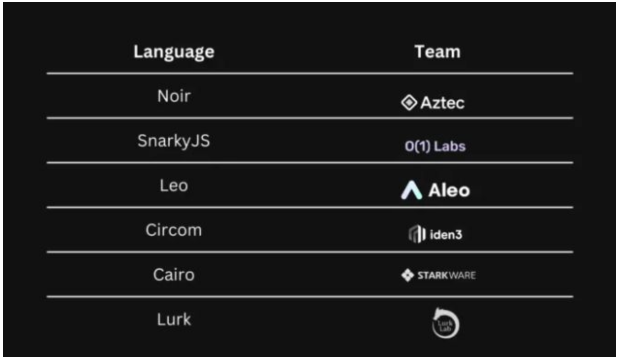
Imperative / description / circuit languages

Different base programming language

Different programming language and framework integration modules

Technological life cycle: all are early stage, but:

- **Successful productive usage**
- **Stable releases**



Language	Team
Noir	Aztec
SnarkyJS	O(1) Labs
Leo	Aleo
Circom	iden3
Cairo	STARKWARE
Lurk	Lurk Lab

Noir concepts

Rust like syntax

Limited cryptographic experience

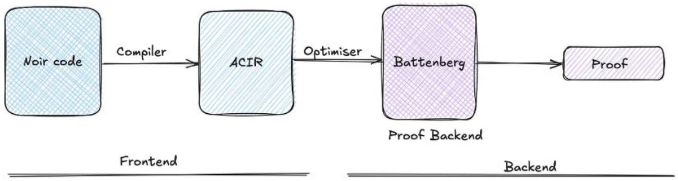
Intermediate representation (ACIR)

Different backends

Platform agnostic, abstract circuit

Different poving backends:

Barretenberg,coSNARKs,Edge,Plonky2,Groth16,



https://github.com/noir-lang/awesome-noir/?tab=readme-ov-file#proving-backends

ACIR

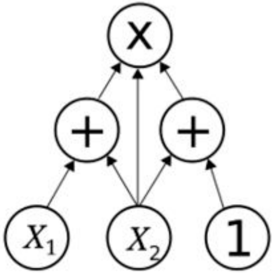
ACIR: abstract circuit intermediate representation

Abstract circuit:

- **DAG (Directed Acyclic Graph)**
- **Gates: opcode - arithmetic constraints**
- **Wires: partial wires**
- **e.g. circuit:** *output_wire = input_wire_1 + input_wire_2*
- **arithmetic constraint:** *output_wire - (input_wire_1 + input_wire_2) = 0*

Intermediate representation: to different backend provers

- user program -> (compilation) ACIR, a list of opcodes which constrain (partial) witnesses
- user inputs + ACIR -> (execution/solving) assign values to all the (partial) witnesses
- witness assignment + ACIR -> (proving system) proof
- blackbox functions



ACIR: https://lib.rs/crates/acir

Noir programming

Data types: Fields, integers, booleans, complex types, public, private

Functions: rust style functions, structs + methods, lambda

Control Flow: control structures, loops

Logical operations: all

Assert: Noir specific predicate

Unconstrained functions: no constraints are generated

Oracles: experimental, unconstrained

Global variables: globally accessible

Lambdas: anonym functions

Noir documentation https://noir-lang.org/docs

Aztec protocol

L2 rollup system on ethereum

Private function possibility

Executing on user device

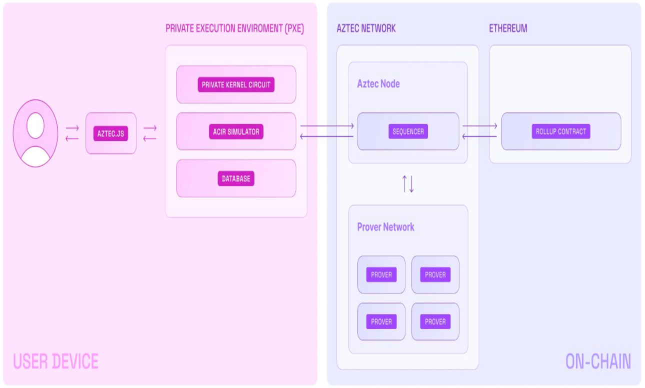
Public and private state

UTXO Ledger - private

Account balance based ledger - public

Noir as a core implementation language

Sandbox for developers



Aztec https://aztec.network/

Aztec sandbox: https://docs.aztec.network/developers/getting_started

Aztec demo

Install sandbox

Start sandbox

Steup:

- **Import accounts**
- **Create account**
- **Deploy contracts**

Mint public tokens

Move tokens to public

Move tokens to private

Aztec sandbox:
https://docs.aztec.network/developers/getting_started

Challenge

Developer challenge:

Experiment with the Aztec sandbox environment on your own

Links, Resources, Literature

Noir : Beginner's Guide I:
https://coinsbench.com/noir-begineers-guide-1-ca43da4f23dd

Noir : Beginner's Guide II
https://coinsbench.com/noir-beginners-guide-ii-188868aa161d

Noir documentation
https://noir-lang.org/docs

ACIR documentation
https://lib.rs/crates/acir

Awesome Noir:
(repository with a lot of examples)
https://github.com/noir-lang/awesome-noir/?tab=readme-ov-file
#proving-backends

Happy Hunting for the SNARK :)

Q & A

Daniel Szego

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