Anatomy of a folding scheme

Sonobe, experimental folding schemes library implemented jointly by 0xPARC and PSE.

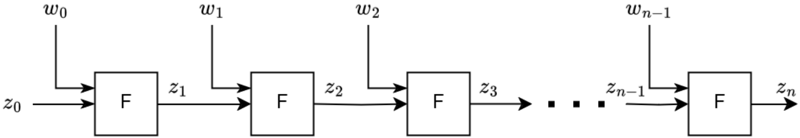
2024-04-22
Barcelona zkDay

Why folding

- Repetitive computations take big circuits $\longrightarrow$ large proving time
  - ie. prove a chain of 10k sha256 hashes
- Traditional recursion: verify (in-circuit) a proof of the correct execution of the same circuit for the previous input
  - issue: in-circuit proof verification is expensive (constraints)
    - ie. verify a Groth16 proof inside a R1CS circuit

IVC - Incremental Verifiable Computation

Folding schemes efficitently achieve IVC, where the prover recursively proves the correct execution of the incremental computations.



In other words, it allows to prove efficiently that
$z_n = F(\ldots\ F(F(F(F(z_0, w_0), w_1), w_2), \ldots), w_{n-1})$.

Folding idea

Homomorphic commitments and RLC

We rely on homomorphic commitments
ie. Pedersen commitments
Let $g \in \mathbb{G}^n,\ v \in \mathbb{F}_r^n$,

$$Com(v) = \langle g, v \rangle = g_1 \cdot v_1 + g_2 \cdot v_2 + \ldots + g_n \cdot v_n$$

RLC:
Let $v_1, v_2 \in \mathbb{F}_r^n$, set $cm_1 = Com(v_1),\ cm_2 = Com(v_2)$.
then,

$$v_3 = v_1 + r \cdot v_2$$
$$cm_3 = cm_1 + r \cdot cm_2$$

so that

$$cm_3 = Com(v_3)$$

Relaxed R1CS

R1CS instance: $(\{A, B, C\} \in \mathbb{F}^{n \times n},\ io,\ n,\ l)$, such that for $z = (io \in \mathbb{F}^l, 1, w \in \mathbb{F}^{n-l-1}) \in \mathbb{F}^n$,

$$Az \circ Bz = Cz$$

Relaxed R1CS:

$$Az \circ Bz = uCz + E$$

for $u \in \mathbb{F}, \quad E \in \mathbb{F}^n$.

Committed Relaxed R1CS instance: $CI = (\overline{E}, u, \overline{W}, x)$
Witness of the instance: $WI = (E, W)$

(We don't have time for it now, but there is a simple reasoning for the RelaxedR1CS usage explained in Nova paper)

NIFS - Non Interactive Folding Scheme

$$CI_1 = (\overline{E}_1 \in \mathbb{G}, u_1 \in \mathbb{F}, \overline{W}_1 \in \mathbb{G}, x_1 \in \mathbb{F}^n) \qquad WI_1 = (E_1 \in \mathbb{F}^n, W_1 \in \mathbb{F}^n)$$
$$CI_2 = (\overline{E}_2, u_2, \overline{W}_2, x_2) \qquad WI_2 = (E_2, W_2)$$

where $\overline{V} = Com(V)$

$$T = Az_1 \circ Bz_1 + Az_2 \circ Bz_2 - u_1 Cz_1 - u_2 Cz_2$$
$$\overline{T} = Com(T)$$

NIFS.P

$$E = E_1 + r \cdot T + r^2 \cdot E_2$$
$$W = W_1 + r \cdot W$$

NIFS.V

$$\overline{E} = \overline{E}_1 + r \cdot \overline{T} + r^2 \cdot \overline{E}_2$$
$$u = u_1 + r \cdot u_2$$
$$\overline{W} = \overline{W}_1 + r \cdot \overline{W}$$
$$x = x_1 + r \cdot x_2$$

New folded Committed Instance: $(\overline{E}, u, \overline{W}, x)$
New folded witness: (E, W)

IVC

U_i: committed instance for the correct execution of invocations $1, \ldots, i-1$ of F'
u_i: committed instance for the correct execution of invocation i of F'

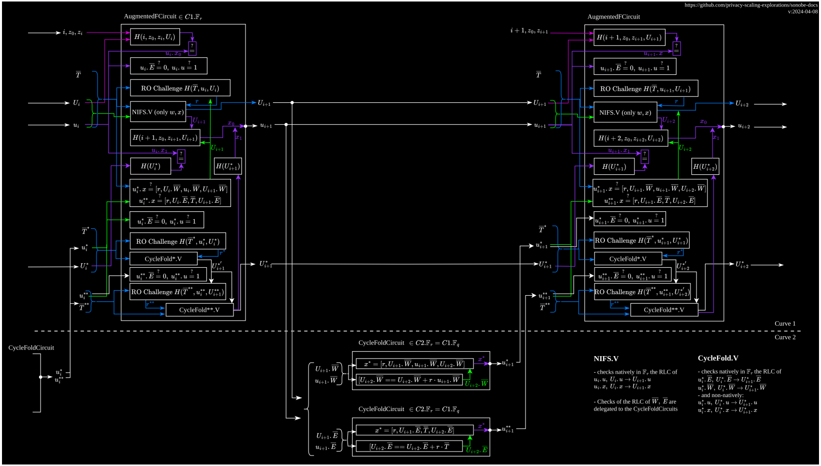
F':
i) execute a step of the incremental computation, $z_{i+1} = F(z_i)$
ii) invoke the NIFS.V to fold U_i, u_i into U_{i+1}
iii) other checks to ensure that the IVC is done properly

Cycle of curves

NIFS.V involves $\mathbb{G}$ point scalar mults, which are not native over $\mathbb{F}_r$.
$\longrightarrow$ delegate them into a circuit over a 2nd curve.

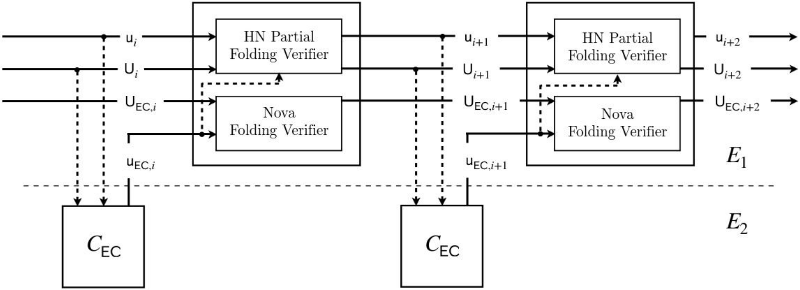
We 'mirror' the main F' circuit into the 2nd curve
each circuit computes natively the point operations of the other curve

Augmented F Circuit + CycleFold Circuit



Other Folding Schemes

Decider



With Prover knowing the respective witnesses for $U_n, u_n, U_{EC,n}$

Issue: IVC proof is not succinct

Decider

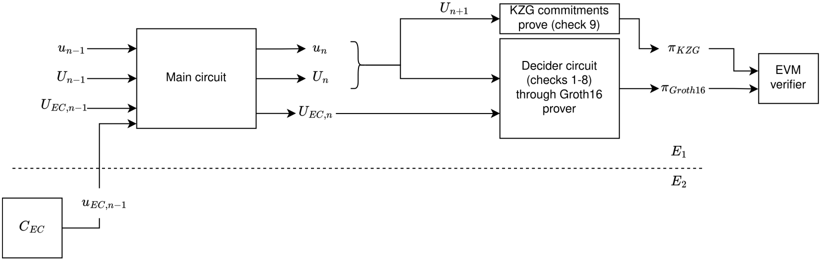
Original Nova: generate a zkSNARK proof with Spartan for $U_n, u_n, U_{EC,n}$
$\longrightarrow$ 2 Spartan proofs, one on each curve (with CycleFold is 1 Spartan proof)
(not EVM-friendly)

Decider

checks (simplified)

1. (U_{n+1}, W_{n+1}) satisfy Relaxed R1CS relation of AugmentedFCircuit
2. verify commitments of $U_{n+1}.\{\overline{E}, \overline{W}\}$ w.r.t. $W_{n+1}.\{E, W\}$
3. $(U_{EC,n}, W_{EC,n})$ satisfy Relaxed R1CS relation of CycleFoldCircuit
4. verify commitments of $U_{EC,n}.\{\overline{E}, \overline{W}\}$ w.r.t. $W_{EC,n}.\{E, W\}$
5. $u_n.E == 0,\ u_n.u == 1$, ie. u_n is a fresh not-relaxed instance
6. $u_n.x_0 == H(n, z_0, z_n, U_n)$
   $u_n.x_1 == H(U_{EC,n})$
7. $NIFS.V(U_n, u_n) == U_{n+1}$

Decider



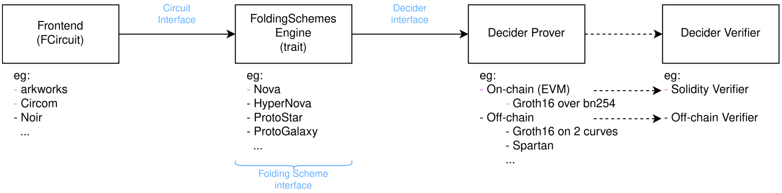
U_{n+1}
KZG commitments prove (check 9)
u_{n-1}
U_{n-1}
$U_{EC,n-1}$
Main circuit
u_n
U_n
$U_{EC,n}$
Decider circuit (checks 1-8) through Groth16 prover
π_{KZG}
$\pi_{Groth16}$
EVM verifier
E_1
E_2
$u_{EC,n-1}$
C_{EC}

Sonobe

Experimental folding schemes library implemented jointly by 0xPARC and PSE.

Dev flow:

1. Define a circuit to be folded
2. Set which folding scheme to be used (eg. Nova with CycleFold)
3. Set a final decider to generate the final proof (eg. Spartan over Pasta curves)
4. Generate the the decider verifier



Code example

[show code with a live demo]

Some numbers (still optimizations pending):

- AugmentedFCircuit: $\sim 80k$ R1CS constraints
- DeciderEthCircuit: $\sim 9.6M$ R1CS constraints
  - < 3 minutes in a 32GB RAM 16 core laptop
- gas costs (DeciderEthCircuit proof): $\sim 800k$ gas
  - mostly from G16, KZG10, public inputs processing
  - will be reduced by hashing the public inputs
  - expect to get it down to $< 600k$ gas.

Recall, this proof is proving that applying n times the function F (the circuit that we're folding) to an initial state z_0 results in the state z_n.
In Srinath Setty words, you can prove practically unbounded computation onchain by 800k gas (and soon $< 600k$).

Wrappup

- https://github.com/privacy-scaling-explorations/sonobe
- https://privacy-scaling-explorations.github.io/sonobe-docs/

2024-04-22

0xPARC & PSE.