Halo

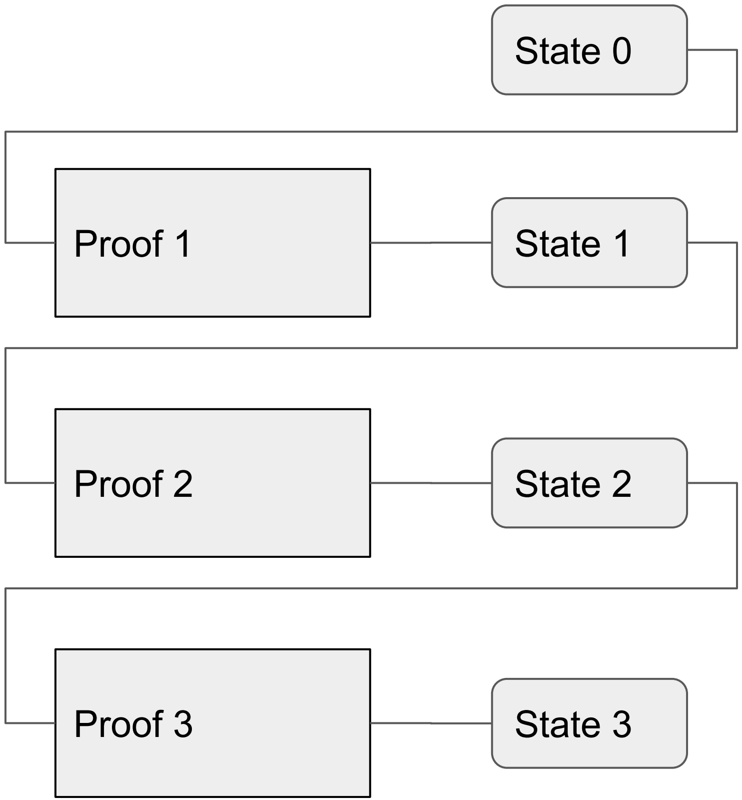
Recursive Proof Composition without a Trusted Setup

Sean Bowe Jack Grigg Daira Hopwood

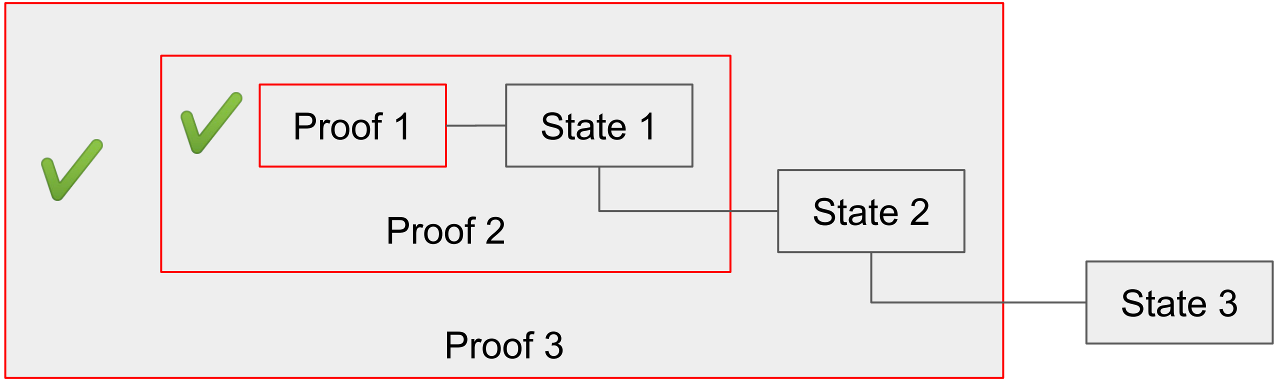
Current Progress

- Pre-print available
  - https://eprint.iacr.org/2019/1021
  - WIP; no security arguments, somewhat out of date
- Implementation
  - https://github.com/ebfull/halo
  - Works! Makes recursive proofs!
  - Slow.

Proof composition



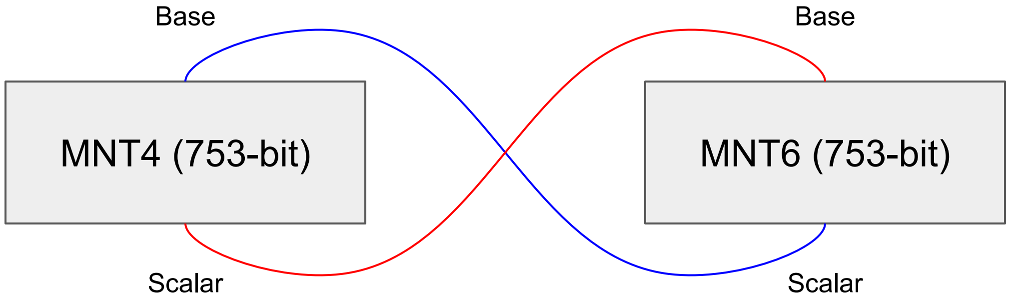
Proof composition



ZEXE (Fixed-depth proof composition)

Pairing-friendly curve “embedded” in another pairing-friendly curve.

BCTV14 (Recursion; arbitrary depth proof composition)



- Keying material is large.
- Multiexps, FFTs, etc. are expensive.
- Needs a trusted setup for each curve, for each circuit.

Universal trusted setup SNARKs

- Updatable and Universal Common Reference Strings with Applications to zk-SNARKs
  - https://eprint.iacr.org/2018/280
- **Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updateable Structured Reference Strings**
  - https://eprint.iacr.org/2019/099
- PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge
  - https://eprint.iacr.org/2019/953
- Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS
  - https://eprint.iacr.org/2019/1047

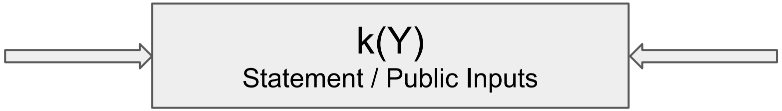
Polynomial commitment schemes

- Lets you commit to a polynomial p(X) of degree at most n.
- Lets you provably evaluate the committed polynomial at arbitrary points.
- Succinct polynomial commitment schemes
  - Commitment is small (usually constant in n)
  - Opening proof is small (usually constant in n)
  - Opening verification is fast (usually constant in n)

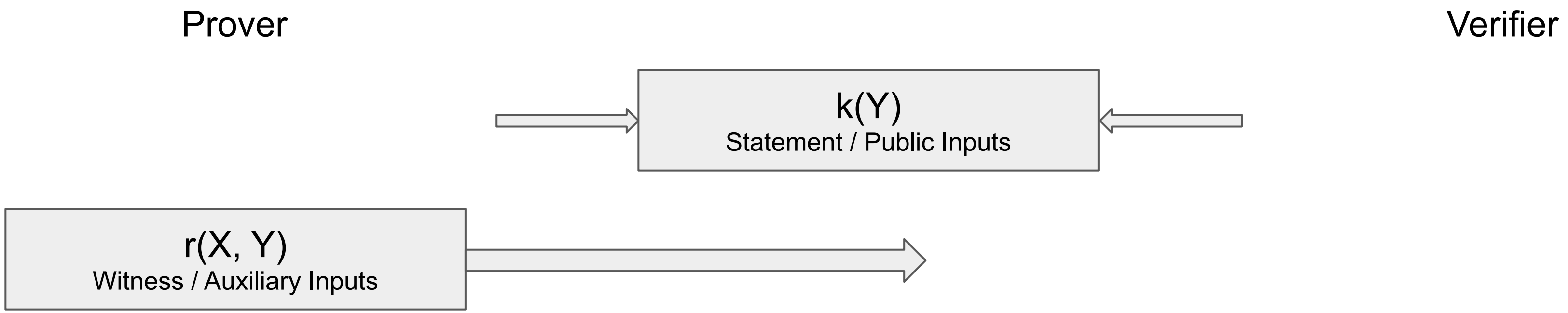
Sonic (core protocol)

Prover

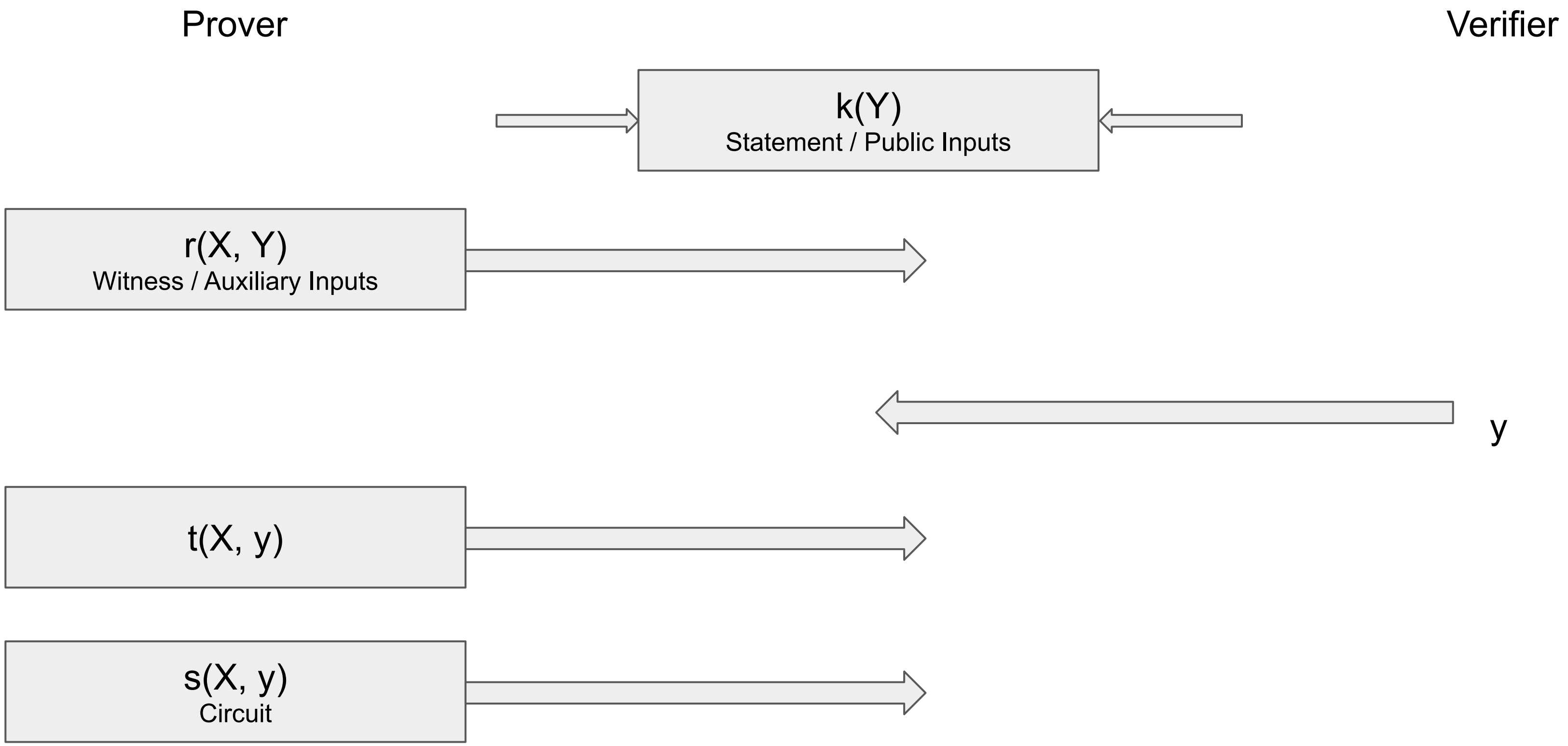
Verifier

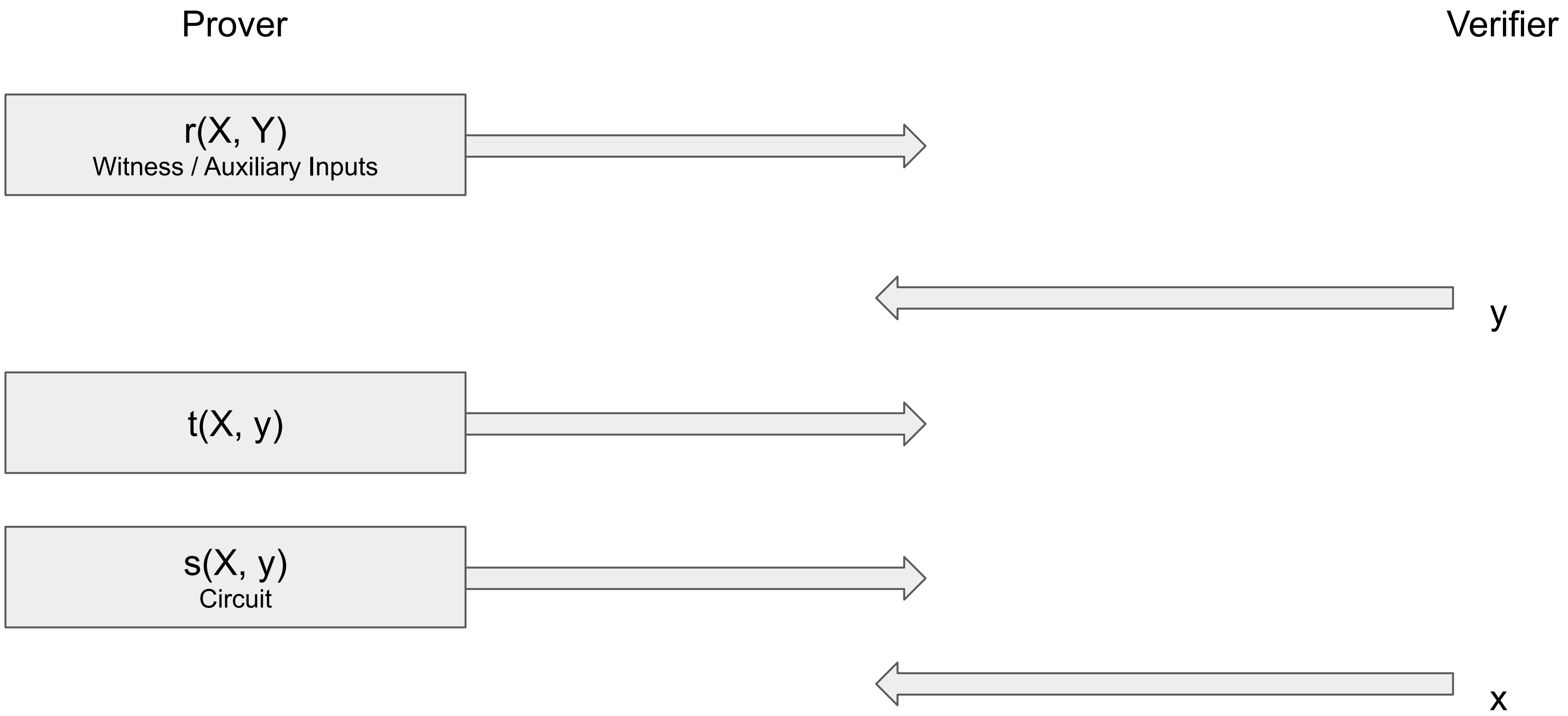


Sonic (core protocol)



Sonic (core protocol)



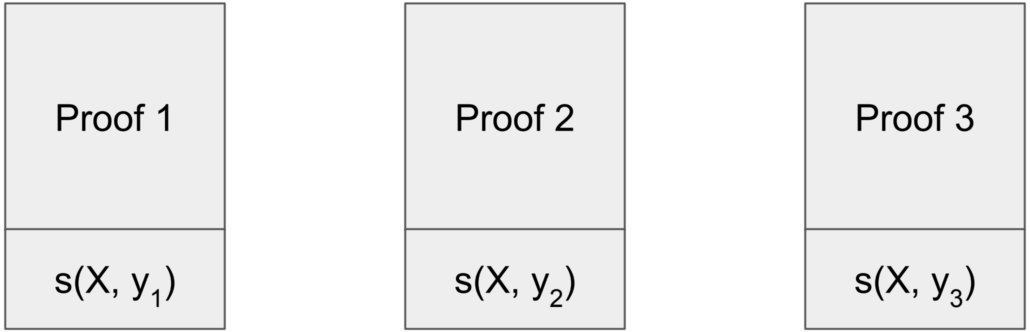


- Is the constraint system satisfied?
    - Check that $t(X, y)$ has a constant term of zero.
    - Check that $r(X, y)$ is degree bound at a particular n
- Is the commitment to $t(X, y)$ correct?
    - $t(x, y) = r(x, 1)\,(r(x, y) + s(x, y)) - k(y)$
- Is the commitment to $s(X, y)$ correct?

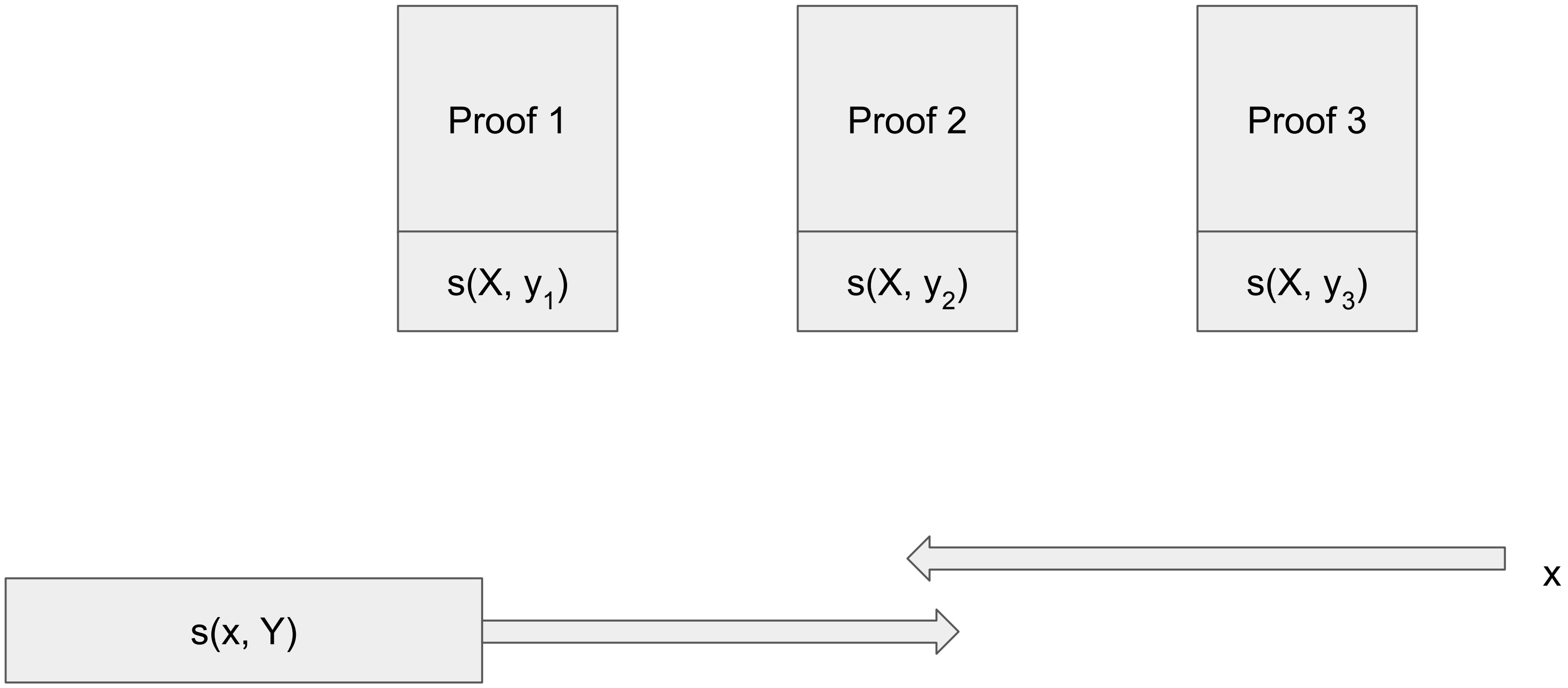
Signature of correct computation

1. Fully succinct check of s(X, y) commitment
2. Amortized succinctness (“helped” mode)

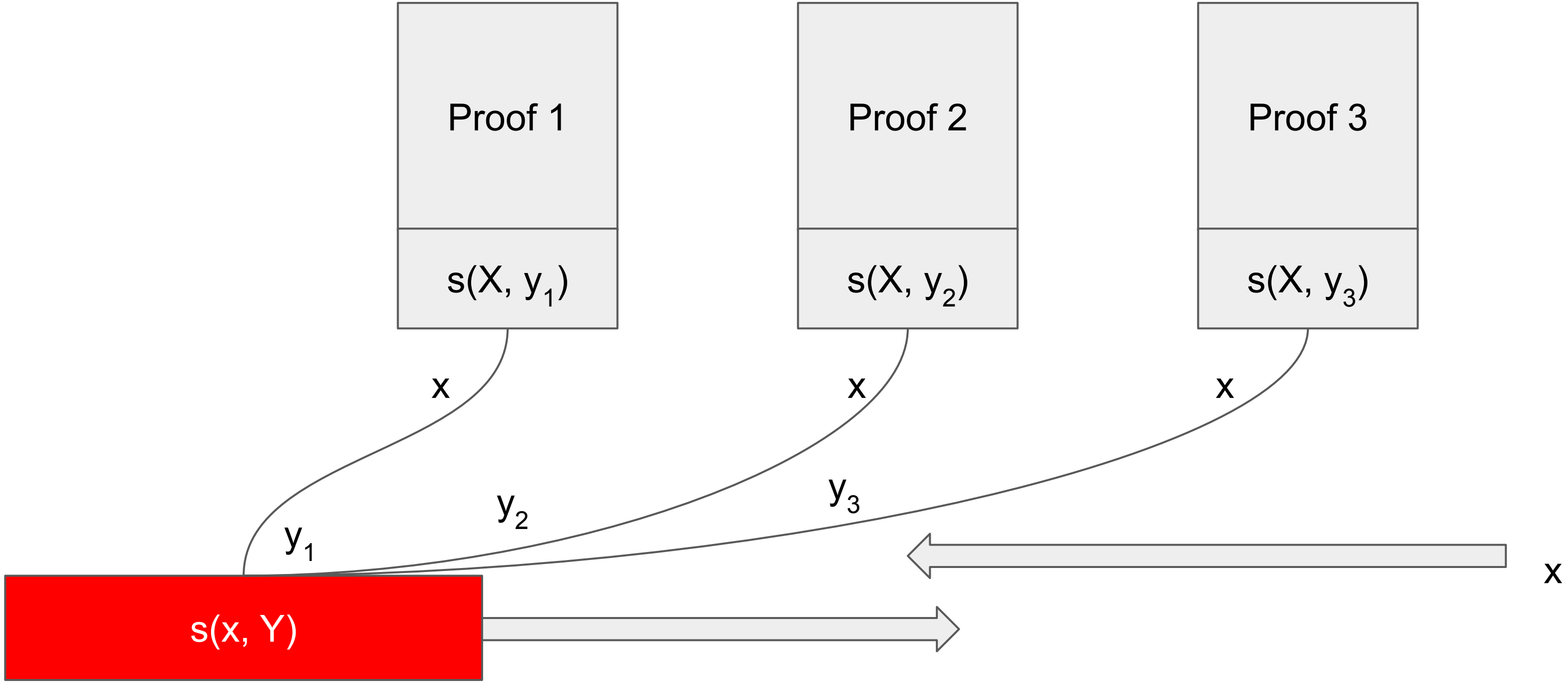
Amortized succinctness



Amortized succinctness



Amortized succinctness



Amortized succinctness

Proof 1

$v_1 =? f(x_1)$

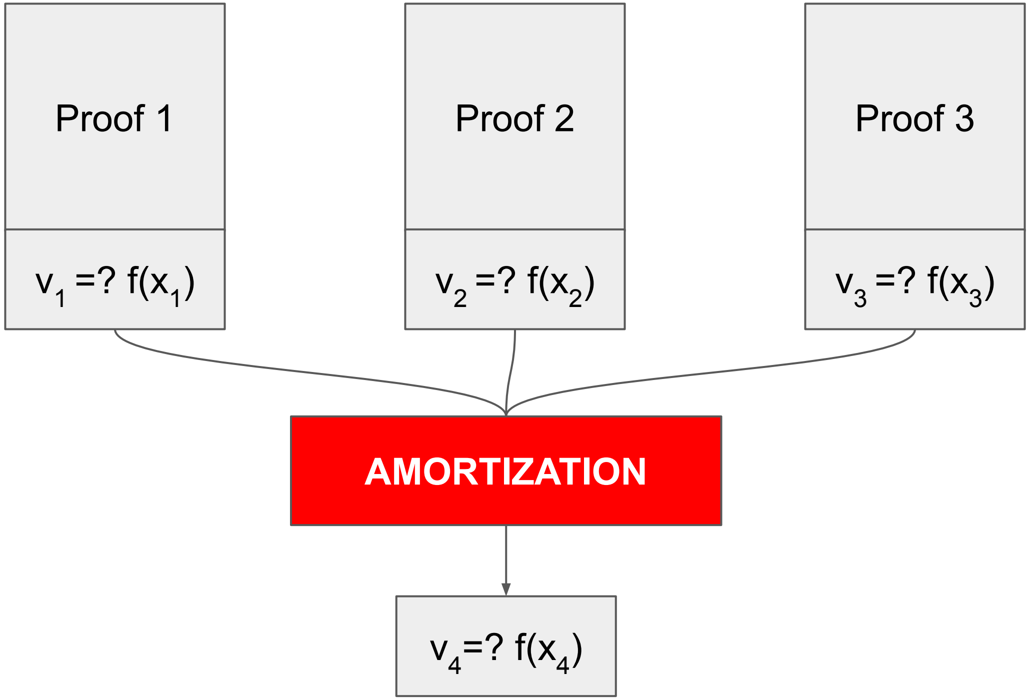
Proof 2

$v_2 =? f(x_2)$

Proof 3

$v_3 =? f(x_3)$

Amortized succinctness

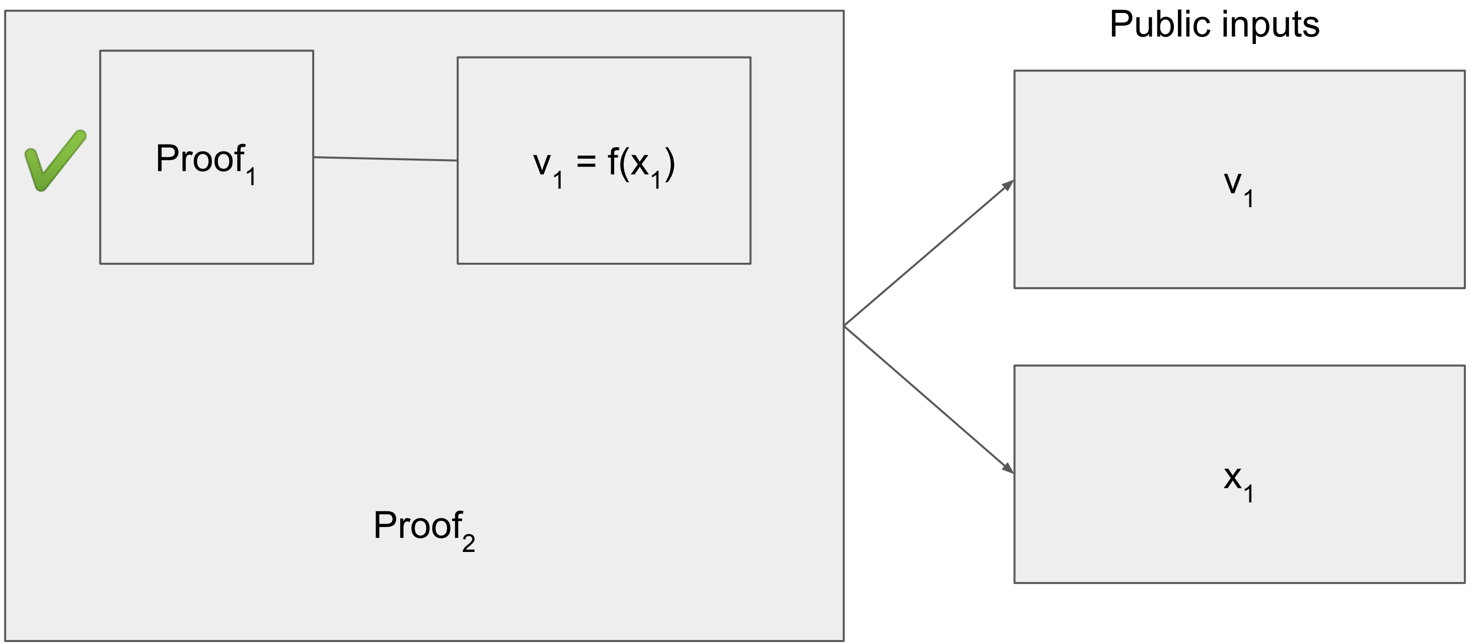


Nested amortization

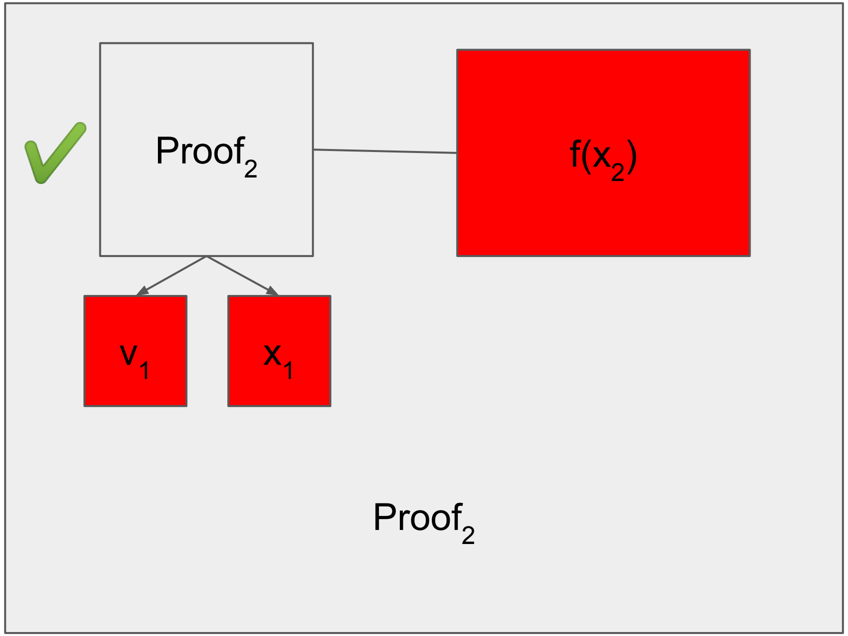
Proof$_1$

In order to verify Proof$_1$, we need the output of an expensive operation $f(x_1)$.

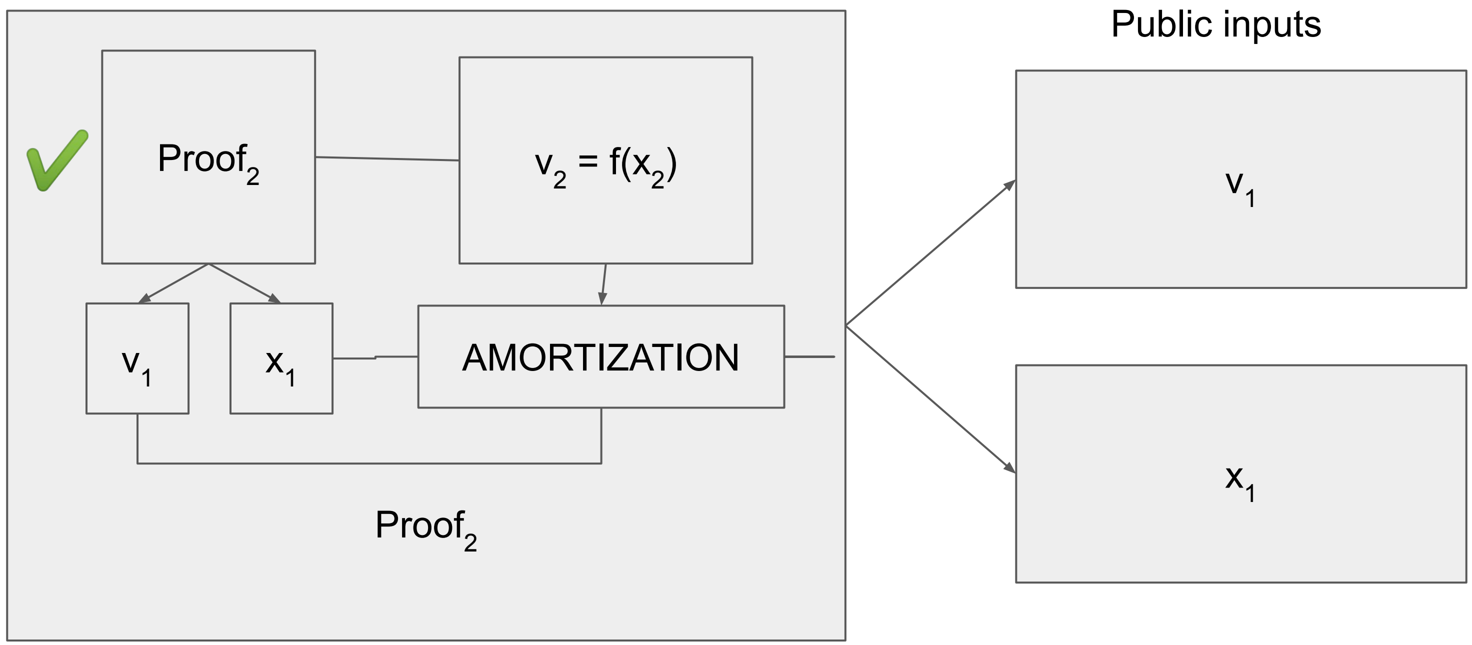
Nested amortization



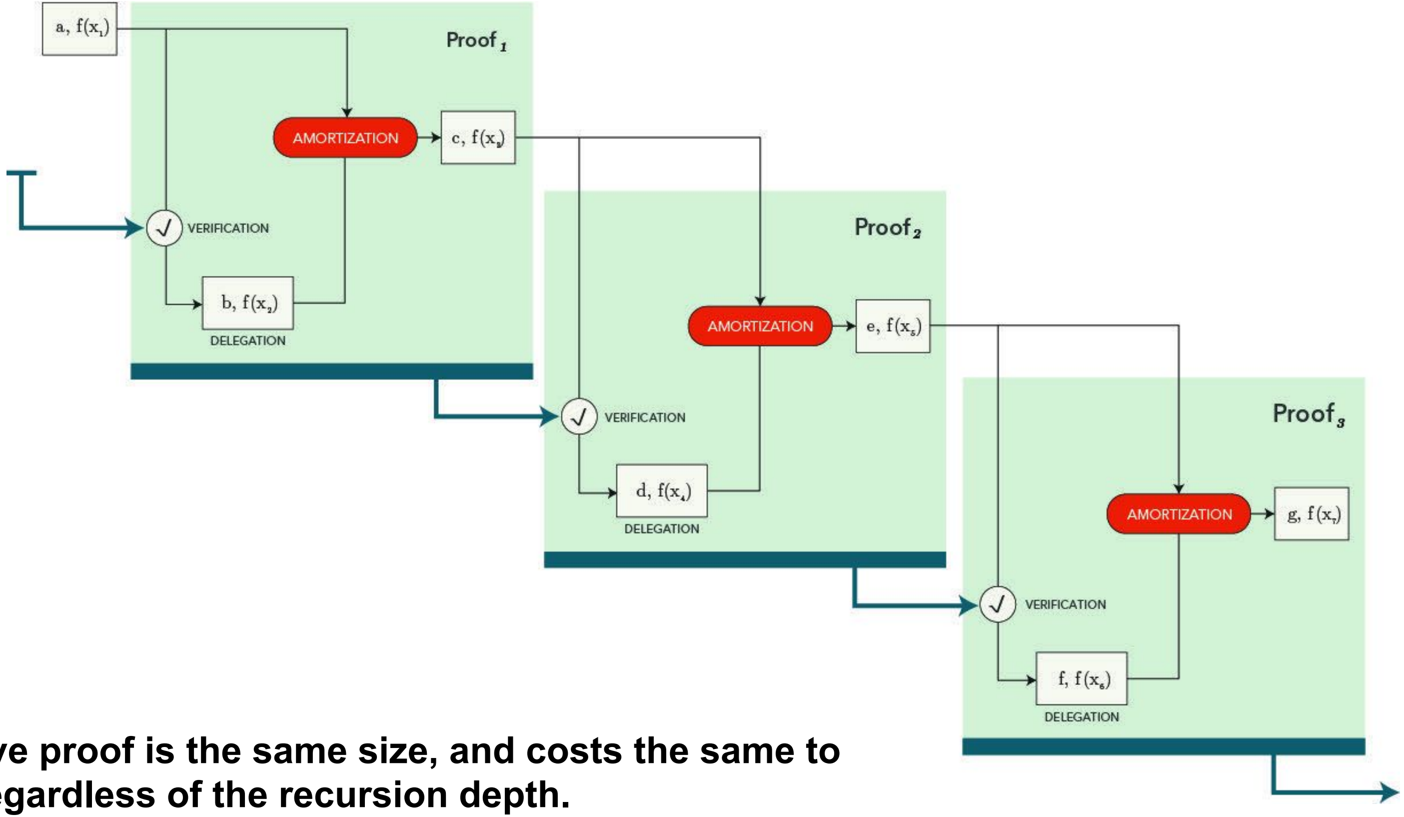
Nested amortization



Nested amortization

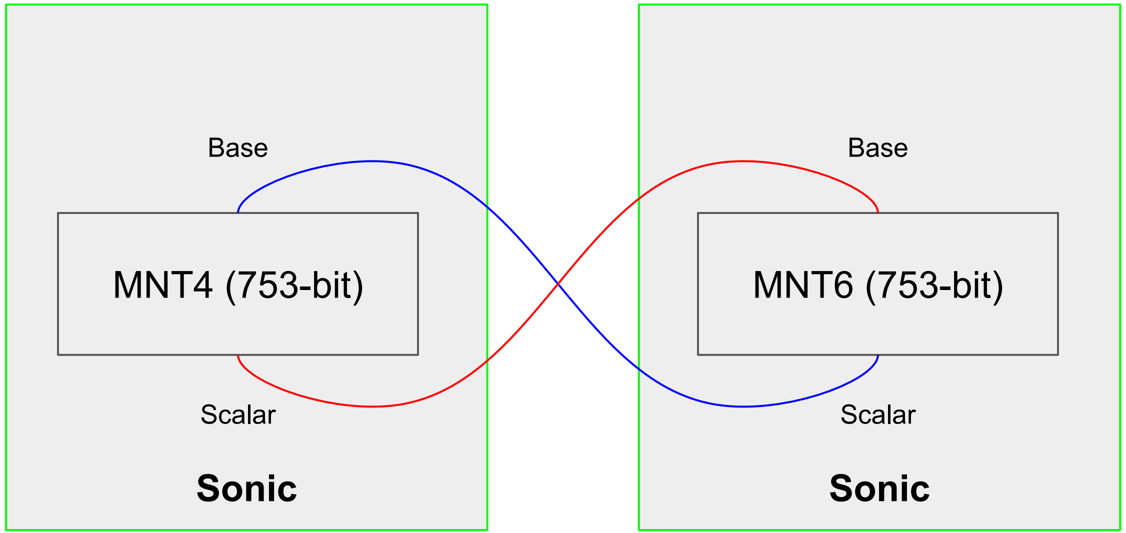


Nested amortization

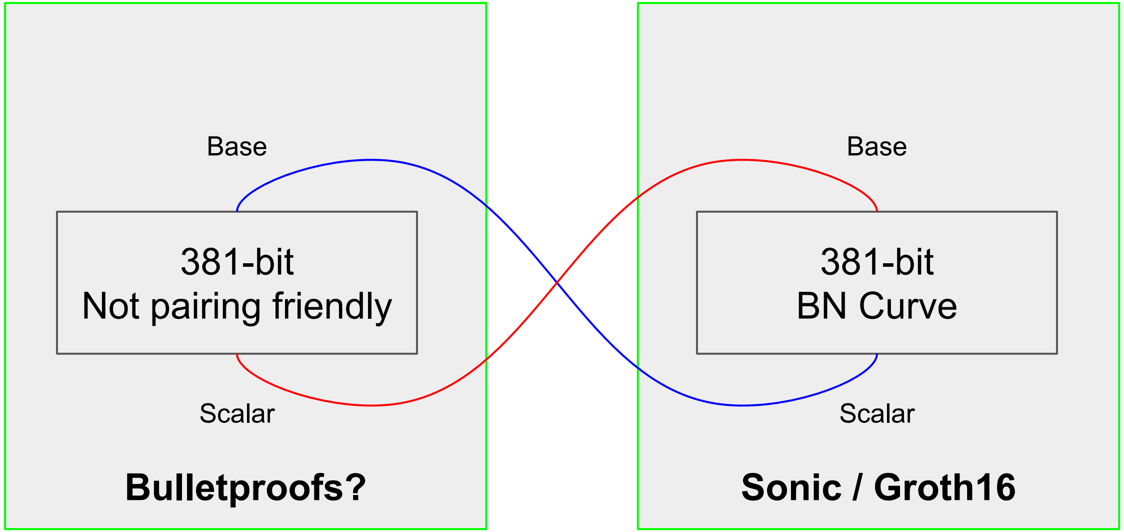


Recursive proof is the same size, and costs the same to verify, regardless of the recursion depth.

Improved recursion



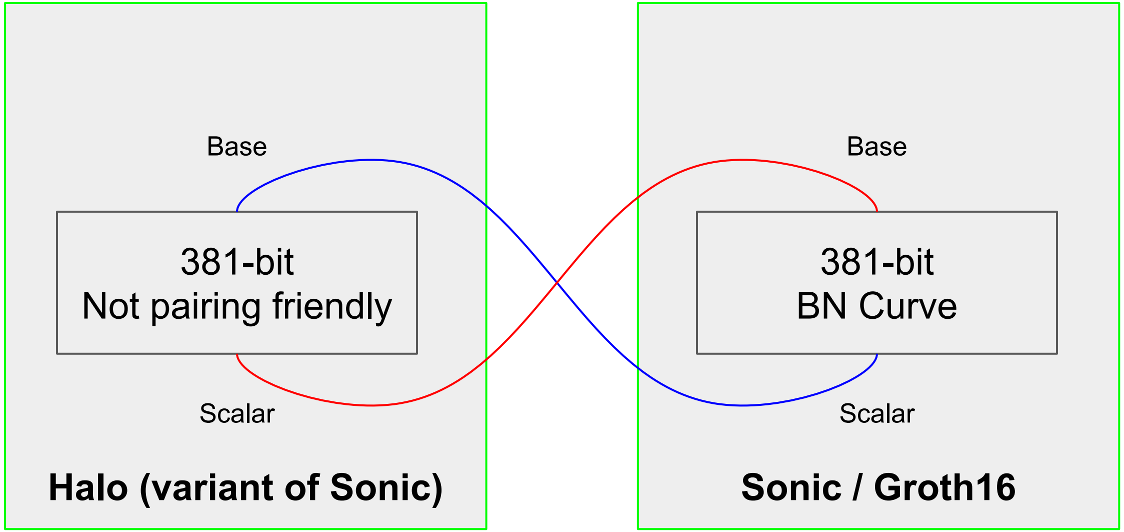
Improved recursion



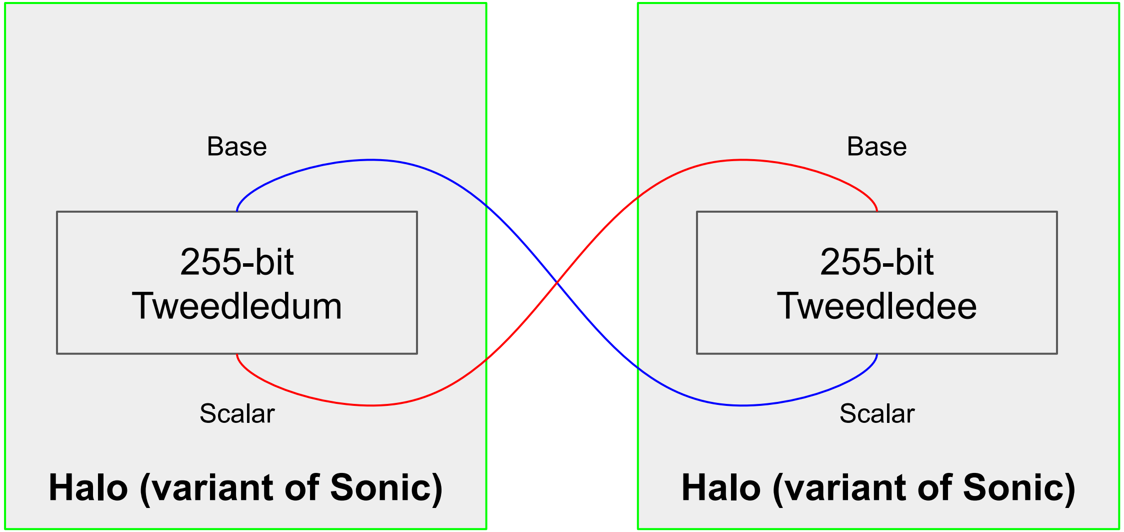
Halo: Sonic on top of the Inner Product Argument

- The inner product argument from Bulletproofs can be used to build a polynomial commitment scheme with all of the properties needed by Sonic.
  - In fact, a zero-knowledge variant of it is available in the Hyrax paper. (https://eprint.iacr.org/2017/1132)
- Crucially, this subprotocol is amenable to an amortization strategy that is identical to the “helped” verifier of Sonic.
- Thus, we can build the “helped” version of Sonic without a trusted setup.
  - Requires some modifications to account for the new commitment scheme.

Improved recursion



Halo: Recursive Proofs without a Trusted Setup!



Halo (more optimizations)

- Delegated work / Interaction between proving systems
  - Reduces size of verification circuit by avoiding non-native field operations
- Smaller challenge spaces
  - Vastly reduces size of verification circuit
- Faster point multiplication using endomorphism
  - Vastly reduces size of verification circuit.
- Faster Pedersen hashes using endomorphism
  - Improves k(Y) commitment evaluation.
- Hashing to squares in the inner product argument
  - Avoids needless squaring at the cost of allowing the prover to fork the transcript
- Collapsing all inner product arguments together into a single argument
  - Uses Kate et al.’s multi-commitment and multi-point opening tricks

Conclusions

- Halo achieves recursive proof composition for the first time!
- Nested amortization technique can be used in conjunction with other strategies
  - Avoiding MNT curves by using half-pairing cycles.
  - Could be used alongside e.g. Fractal so that end-user proof size is small.
- Inner product argument batching technique can be used to speed up Bulletproofs.
- Faster point multiplication tricks can be used to improve verification of public coin interactive protocols inside of circuits.
- Faster Pedersen hashes
- Interactive strategies between proving systems could be used to address performance problems in future protocols
- Thanks!