ELVES: Efficient block-auditing for blockchains

Aka Polkadot's sharding since 2021

Researchers:

Jeff Burdges[†] Alfonso Cevallos Handan Kılınç Alper
Chen-Da Liu-Zhang[†] Fatemeh Shirazi Alistair Stewart

Implementors:

Alex Gheorghe Rob Habermeier Robert Klotzner
Bastian Köcher Andronik Ordian Maciej Kris Żyszkiewicz
Andrei Sandu

9 August 2024

https://eprint.iacr.org/2024/961

All of web2 requires “sharding” aka horizontal data partitioning,

real services grow beyond what any one machine can do.

Blockchains simulate a database through auditing!

Are blockchains only tools for costly “toy” services?

Or can the auditing be made more efficent?

Block-auditing protocol questions:

Is it provably secure?

Under what assumptions?

How must does it cost?

Computation? Bandwidth? Oportunity? Energy?

Units of auditors/verifiers

Overall protocol questions:

Censorship resistance? Liveness? Storage costs?

Idea: Bridge network

Disadvantage: Adds another 2/3rd honesty assumption per shard

Idea: Optimistic roll ups

Not provably secure, or else
wild assumptions like under-explained validation

Disadvantages: Inter-shard latency of a week

Idea: Cryptographic proofs aka zk roll ups

All blocks have some succinct-ish cryptographic proofs..

Advantages:

- Weakest assumptions, but exactly what depends.

- Recursive in theory. And composable.
  Verify ancestry ala Mina

“At present zk roll up provers cost 1 to 100 million times the CPU cost of a single verifier” - Justin Thaler (a16z) 2023 paraphrased

Disadvantages:

- High computational cost

  ETH pays over half a million nodes, but elsewhere?

- Inter-shard latency in hours, likely worsens MEV.

- Oportunity costs of decentralized provers.

- And specialized hardware improves direct verifiers similarly.

Idea: True sharding aka select committee in advance

Assume high honesty threshold like 80%
plus unbiased randomness plus . . .

Argue 1000 random validators should be 2/3rds honest,
using some concentration inequality like Hoeffding or

Theorem (Chebyshev's inequality)

$$\Pr(|X - \mathrm{E}[X]| \geq a) \leq \frac{\mathrm{Var}[X]}{a^2}$$

OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding by Bryan Ford's DEDIS group (EPFL)

Idea: True sharding via concentration inequalities

- 1000 x CPU $\ll$ 1 million *but* 1000 x bandwidth sucks.

Advantages:

- Does off-chain protocols too, not just block auditing.
  And stateful too.
- Inter-shard latency in seconds or minutes, post-finality
  Aka bridges become fast & trustworthy

Pseudo-disadvantages:

- Non-recursive, but not necessary
- Few if any apply correctly

  Even "professor coins" have shards too small, e.g. dFinity 15-40

Tool: Soundness slashing

Some slashing improves Byzantine agreement anyways but..
very messy, few rigorous results, silly claims (PoW lol)

"Professor coins" do not slash, e.g. Avalanche, Cardano, dFinity

Soundness slashing is simpler: 100% for provable violation

Assume total stake is $\nu = 5\%$ of the value M of the network, then expected profit becomes

$$\varepsilon M - (1 - \varepsilon)\nu M/n < 1.$$

Ask soundness error $\varepsilon < (20n)^{-1} = 20001^{-1}$

Tool: Stateless blocks

Side/para-chain makes block,
attaches state roots and proofs for state reads & writes.

Merkle proofs multiplies `block_size` by $32 \log$ `state_entries`

Alternatives:

KZG seems too slow. Aggregate binary KZG?
Lattice-fold of lattice hashes? Stwo?

Computation vs bandwith & opertunity trade-off here

Tool: Availability commitment via erasure coding

Block body becomes code word, divided into chunks/symbols.
Add their Merkle root to block header.

Are the chunks retrievable?   Is encoding correct?

2d Reed-Solomon:

Attacker must hide 1/4th, so sample lots!

Allows partial reconstruction for $O(\sqrt{n})$ sized fraud proofs,
but who cares? Validity is not a property of the whole block.

Tool: Availability commitment via erasure coding

Block body becomes code word, divided into chunks/symbols.
Add their Merkle root to block header.

Are the chunks retrievable? Is encoding correct?

2d Reed-Solomon: REJECTED

1d Reed-Solomon:

Attacker must hide 2/3rds, so assuming 2/3rd honest suffices.

One chunk per node, so bandwidth is only 3 downloads.

All or nothing reconstruction, so reencode when checking blocks,
and allows batch verification etc in blocks.

Idea: Byzantine “cut n choose” proof

Assume 2/3rd honest plus “some” synchrony. Also slashing.

side/para-chain:

0. Make state-less blocks

hot beacon/relay chain:

1. Declares responsibility
   Some validator receives, checks, and “backs” the blocks
2. Prove checkability using 1d Reed-Solomon
   Relay chain wait upon 2/3rd availability votes
3. Inter-shard messages sent optimistically

. . .

Idea: Byzantine "cut n choose" proof

Assume 2/3rd honest plus "some" synchrony. Also slashing.

. . .

hot off-chain:

4. Validators self-assign in approval gadget
5. Finality gadget waits upon approval gadget proves validity

cold off-chain:

6. Escalate if anyone disputes in approval gadget
7. Revert relay chain & kill backer if dispute succeeds

cold/tomorrow on-chain:

8. Accumulate rewards votes

Advantages:

- Inter-shard latency of 10s of seconds, even pre-finality
- 30-35 checkes, 29x cheaper than OmniLedger.
  15-20 may be possible!
- Prove security by exhausting stake!
- Pipelines & batches well

Disadvantages:

- Need pipelining & batching
- Complexity, probably no worse than other pipelined
- 100% slashing
- Soundness assumes 2/3rd honest plus "some" synchrony
- Gossip is $\Omega(n^2)$ bandwidth,
  but 1000 works, so composes with true sharding

Auditing tasks in ELVES

Endorsing Light Validity Evaluator System

0. Announce/gossip your local VRF that assignes you.
   synchrony!
1. Reconstruct from availability
2. Reencode to check erasure root
3. Check stateless block
4. Announce/gossip the result

Idea behind ELVES approval gadget

Endorsing Light Validity Evaluator System

One of two VRFs sampling modes assign auditing tasks c

- Tranche zero samples wide across jobs

  $(c_1, \ldots, c_\mu) \leftarrow \mathsf{VRF}_V(r) \bmod \texttt{num\_cores}$

  where $\mu\, n \approx s_0\, \texttt{num\_cores}$ and $s_0 \approx 30$

- Tranche $t_c > 0$ samples a delays, usually never reached.

  $\mathtt{t_c} \leftarrow \mathsf{VRF}_V(r \mathbin{+\!\!+} c) \bmod \lfloor n/s_\delta \rfloor$ with $s_\delta \approx 2.25$

Another tranche goes for each assignees who "no shows"
aka goes 12 seconds without voting.

Everyone approves, disputes, or no-shows eventually.

Non-zero tranche width $s_\delta \approx 2.25$?

If $s_\delta \leq 1.5$ adversaries can DoS honest asignees.

If $s_\delta \geq 3$ too few adversaries can DoS network.

γ	s_δ	s_f / $\ln(1/\varepsilon)$
$1/5$	2.3105	2.2737
$1/4$	2.1972	4.4376
$1/3$	2.0794	9.7767

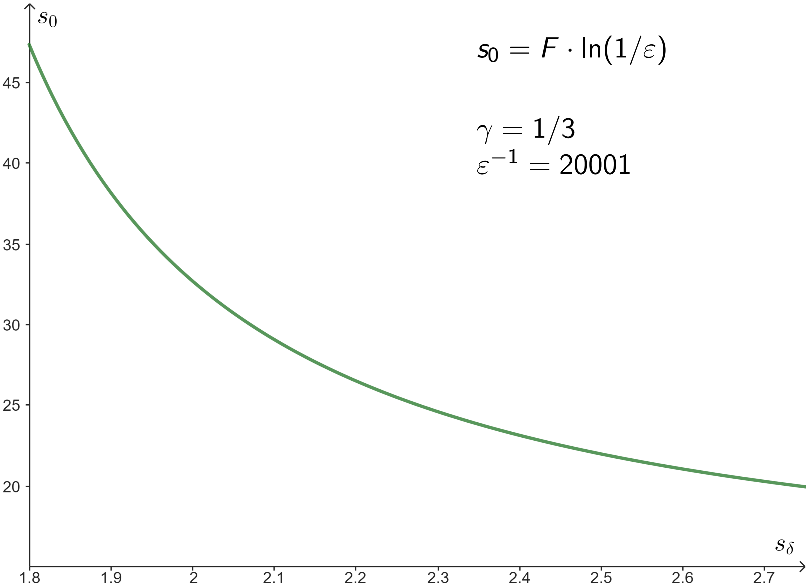
$$\ln(1/\varepsilon) \approx 9.903538 < 10$$

Zeroth tranche width $s_0 \geq F \cdot \ln(1/\varepsilon) \approx 25.5$ where

$$F := \frac{\mathsf{s}_\delta}{\ln\left(\frac{-(1-\gamma)\,\mathsf{s}_\delta}{W_0\left[-(1-\gamma)\,\mathsf{s}_\delta \cdot e^{-(1-\gamma)\,\mathsf{s}_\delta}\right]}\right)} \approx 2.57373251367$$

Alternatively:

Run simulations where adversary silences all honest assignees



$$s_0 = F \cdot \ln(1/\varepsilon)$$

$$\gamma = 1/3$$
$$\varepsilon^{-1} = 20001$$

Polkadot/Kusama performance:

6 second blocktimes. Parachain can run faster via elastic scaling.

Parablocks have 5 mb blocks including state proofs, and 2s of single threaded WASM execution. PolkaVM is better.

We've run ELVES since 2021 but slowly addressing bottle necks.

At present Kusama has 100 "cores" on 1000 validators, but 200 looks ready, so that's 1 blockcain per 5 validators, and 1 blockchain every 3 validators looks doable. Two? Parity?

We force upgrades of software, but not yet of hardware, so nodes remain potatoes: Recommed four CPU cores, asshats run two.

Off-chain work demands off-chain rewards protocols, designed but not yet implemented. And community matters.

Observation:

ELVES gains dramatically from being off-chain, so no ETH roll up.

Small per block committees maybe useless without 100% slashing.

“No shows” happen. Past escalations were only mostly our fault.

Anti-upgrade ideology makes everything impossible.

“The hammer has dropped. The protocol has indicated that a finalized block be reverted. Please inform an adult.” #1349

All the complexity hides under the rug in this talk:

Approval gadgets are kinda straightforward.

Backing is a nightmare. Benchmark all the things!