Mechanism Design Audit of Crosslink Zebra

Nicolás "nikete" Della Penna

January 2026

Process note. This audit is part of an iterative, collaborative process to strengthen Crosslink Zebra's security design. It reflects the current state of a design that is itself evolving. The findings, failure-mode analyses, and recommendations are intended to surface trade-offs and inform design decisions—not to render a final verdict on the system's security. As the design matures, so should the analysis; this document is meant to be revisited and revised alongside the specification it examines.[1]

Abstract

This report presents a mechanism design audit of Crosslink Zebra, a hybrid proof-of-work / stake-weighted finality mechanism for Zcash. The audit evaluates whether the incentives faced by miners, finalizers, stakers, and the broader community make irreversible finality an equilibrium under realistic adversarial behavior. We analyze Penalty for Failure to Defend (PFD) for safety, finality progress (liveness), and censorship resistance; identify equilibria and catastrophic failure modes; and provide concrete, spec-compatible recommendations. The analysis assumes no slashing and strong privacy constraints, focusing on opportunity-cost penalties and explicit governance backstops.

1 Executive Summary

Crosslink Zebra augments Zcash PoW with stake-weighted Byzantine finality. The core finding of this audit is asymmetric:

- **Safety (irreversibility) is structurally strong** once finality is reached, assuming standard BFT assumptions (e.g., fewer than the tolerated Byzantine threshold) and PoW consistency.
- **Finality progress (liveness) is economically fragile** in the baseline design and requires explicit incentive constraints to avoid rational and entropic failure.

The audit identifies three primary failure modes that merit attention as the design evolves:

1. **The Hostage Holdout:** A blocking coalition rationally delays finality to increase future payouts.
2. **The Zombie Set:** The finality gadget becomes permanently inoperable due to validator attrition during a freeze.

[1] This review was conducted against commit `f1b6bf60aa2313caf37252f4bd9114dd06fa1bc1` of the Crosslink book repository.

3. **The Liquid Exit Paradox (conditional):** If stake can exit economically while the frozen validator set retains voting weight for finality over an extended stall, the effective cost to corrupt the frozen set may decay over time, creating "phantom security." Whether this is possible depends on the staking epoch rules and whether exit actions remain feasible during stalls.

These are structural trade-offs inherent to hybrid consensus designs; surfacing them now, while the specification is still taking shape, creates the opportunity to address them before they become operational constraints.

We propose core changes and governance-level extensions that preserve privacy, avoid slashing, and materially improve liveness robustness.

Core recommendations.

1. **R1: Signer-conditioned commission.** Pay finalizer commission only to keys that sign the canonical finality certificate.
2. **R2: First-inclusion miner bounty.** Pay a bounded, single-use bounty to the first PoW block that advances finality, keyed to *finalized height* to prevent malleability gaming.
3. **R3: Finality-gap telemetry.** Expose the finality gap $G = h - \mathrm{LF}(h)$ and related signals for clients and operators.
4. **R4: Anti-jackpot constraint.** Prevent pending rewards from increasing during stalls via a cap or decay. This is a *liveness requirement*, not a tuning option.

Governance & Structural extensions.

1. **R5: Liveness reset.** Define a governance-level mechanism to reset the active set after prolonged stalls.
2. **R6: Certificate aging.** Apply miner bounties to any certificate that advances finality, optionally weighted by the number of blocks advanced to encourage catch-up.
3. **R7 (Optional): Stall–exit coupling.** Explicitly specify whether and how staking exits can complete during prolonged stalls to avoid security-budget decay of the frozen set.

2 Scope, Assumptions, and Threat Model

In scope. Economic incentives of miners, finalizers, and stakers; finality progress; censorship resistance; and governance backstops.

Out of scope. Cryptographic soundness, node implementation bugs, detailed macroeconomic modeling of ZEC price, and full cryptographic privacy analysis.

Assumptions. We assume economically rational agents capable of coordination, partial synchrony, no slashing of principal stake, and privacy constraints consistent with Zcash's design goals.

Privacy and attribution. Privacy constraints apply to shielded transactions and stake ownership, not to consensus participant attribution. Finalizers are assumed to have identifiable signing keys for the purpose of consensus and reward accounting. Incentive mechanisms must not increase linkability of stake or transaction histories beyond what is already implied by consensus participation.

Irrational griefing. Actors willing to incur unbounded losses to cause disruption cannot be deterred by any incentive mechanism. Such behavior is outside the scope of mechanism design and must be addressed through governance or legal means.

3 Mechanism Overview

Each PoW block carries BFT context referencing a finality certificate. Verifiers compute a last-final function $\mathrm{LF}(\cdot)$. A block updates finality if and only if

$$\mathrm{LF}(\text{child}) > \mathrm{LF}(\text{parent}),$$

where $\mathrm{LF}(\cdot)$ refers to the finalized height referenced by the block's BFT context, not the PoW parent pointer.

Rewards, stakers, and commission. A notable finding of this audit is that the in-scope design computes the staker reward slice for *all stakers* based on total stake weight, *regardless of whether their staking positions are assigned to an active-set finalizer*. In contrast, the commission slice is paid only to finalizers in the active set.

The nuance that total payout sums to S_{total} *or less* arises from the specific mechanics of the commission slice: any commission attributable to stake assigned to non-active finalizers is not paid out (burned or returned to pool). Staker rewards are not reduced by assignment to non-active finalizers.

Validator-set freeze. The active validator set is frozen at the last finalized block. During stalls, redelegation cannot alter the set.

Finalization horizon vs. reorganization limits. Finality certificates must respect a bounded *reorganization* limit: a certificate may not contradict or re-finalize history that has already been finalized beyond a bounded window.

This restriction does *not* prohibit finalizing older unfinalized history after a prolonged stall. Following a stall, the frozen validator set may issue certificates that advance finality forward from the last finalized height toward the current PoW tip ("catch-up finalization"). The bound applies to reverting finalized history, not to finalizing previously unfinalized blocks.

4 Role-by-Role Incentive Analysis

4.1 Miners

Miners choose whether to include a certificate reference. Without an explicit incentive, inclusion may be dominated by strategies that preserve a "loose" tip for MEV extraction or reduce orphan risk.

Miner bounty (R2). A bounded, single-use bounty δR—where $\delta \in (0, 1)$ is a fixed fraction of the pending pool—paid to the first block that advances finality removes miner indifference.

Malleability Risk (Certificate Gaming). BLS multi-signatures allow subset aggregation. If the bounty is paid based on the certificate *hash*, miners can malleate a certificate (e.g., using a subset of signatures) to create a "new" certificate and claim the bounty or orphan competitors. *Correction:* The bounty must be keyed to the **Finalized Height**, not the certificate hash, ensuring it is paid exactly once per advancement regardless of which valid certificate version is included.

Incentive Distortion: The Empty Block Race. A significant side effect of R2 is the "Empty Block" strategy. If the bounty δR is large relative to transaction fees, miners are incentivized to minimize block propagation latency and verification time to win the race.

- *Strategy:* Produce a block containing *only* the finality certificate and the coinbase transaction, stripping all user transactions.
- *Impact:* While finality advances, transaction throughput drops to near-zero during high-value certificate periods.
- *Mitigation:* Calibrate δ and certificate encoding so that (i) certificate verification is lightweight, and (ii) the expected marginal orphan-risk cost from including typical transaction load is small relative to transaction fees.

Interaction with R4. If R4 caps or decays the pending pool during prolonged stalls, the effective miner bounty is also capped or decayed. This weakens miner incentives precisely during extended disruption. Implementations may therefore decouple the miner bounty from the pending pool or rely on governance escalation once incentives flatten.

4.2 Finalizers

If commission is paid regardless of participation, finalizers can free-ride. Conditioning commission on signing (R1) removes this equilibrium but does not prevent coordinated stalls by $\geq 1/3$ coalitions.

Equivocation Spam (DoS Hygiene). In the absence of slashing, equivocating (signing two conflicting certificates for the same height) carries no financial penalty.

- *Attack:* A malicious finalizer (or compromised key) floods the network with conflicting signatures.
- *Impact:* Honest nodes must expend computation and bandwidth verifying signatures that are eventually discarded. This acts as a costless Denial-of-Service (DoS) vector against the P2P layer.
- *Mitigation:* Treat equivocation as a networking-layer abuse vector: implement duplicate-signature suppression, bounded gossip fanout for conflicting votes, and peer-scoring/disconnect policies for repeated equivocation broadcasts. This is not slashing; it is standard DoS hygiene for consensus gossip.

Distributional impact under attack. During a coordinated stall, reward decay applies to the entire pending pool. Honest finalizers lose rewards they would have earned, while attackers—having already withheld participation—may not incur additional marginal cost. This reinforces the need for observability and governance escalation in prolonged attacks.

4.3 Stakers

Stakers discipline operators via delegation only when finality advances. During stalls, market-based repair is disabled. Because staker rewards accrue regardless of assignment to an active-set finalizer, stakers' primary incentives relate to liveness, commission competition, and governance externalities rather than direct reward eligibility.

4.4 Miner–Finalizer Collusion

Entities controlling both mining power and finalizer stake can coordinate to delay finality and extract MEV. Incentives and observability reduce profitability but do not eliminate short-term coordinated disruption.

4.5 Structural Intersections: The Fork Choice Gap

The interaction between the PoW Fork Choice Rule and the Finality Gadget creates a critical structural dependency.

Work vs. Finality Preference. If the Fork Choice Rule prioritizes "Total Work" over "Finality Consistency," miners can effectively bypass a stalled finality gadget by forking from a pre-stall block and ignoring the gadget entirely. This negates the "Hostage Holdout" (miners ignore the hostages) but degrades the system to pure PoW, violating the safety goals.

Recommendation. The protocol must explicitly define the Fork Choice Rule as "Heaviest Chain *that extends the latest known Finality*." This enforces safety but confirms that a gadget stall results in a chain halt (Liveness Failure), necessitating the Governance Reset (R5) mechanism.

5 Penalty for Failure to Defend (PFD)

[PFD] The Penalty for Failure to Defend is the opportunity cost imposed by the protocol on a deviation, relative to compliant behavior.

In a non-slashing design, PFD consists of withheld rewards, delayed liquidity, and operational costs.

5.1 Dynamic ransom

If a coalition stalls finality for Δ blocks and then finalizes once, a simplified payoff is

$$U(\Delta) = \beta^{\Delta} \cdot R(\Delta),$$

where $R(\Delta)$ represents the accumulated reward available after delay Δ. If $R(\Delta)$ grows with Δ and agents are sufficiently patient, delaying finality is optimal.

Discounting vs. hazard. We distinguish time preference β from an exogenous hazard rate (e.g., probability that finality advances or governance intervenes). Both act as effective discounting but represent distinct sources of risk.

5.2 Bribery and pivotality

Stalling does not require bribing a full one-third of stake. If honest participation is $p > 2/3$, an attacker need only bribe enough participants to reduce effective participation below the threshold. The cost to stall therefore depends on the *margin* to 2/3, not total stake.

6 External Incentives and Impossibility

Let agent utility decompose as

$$U_i = U_i^{\text{protocol}} + U_i^{\text{external}},$$

where U_i^{external} captures off-chain positions (e.g., short exposure).

Proposition 1. *In any non-slashing mechanism,* $\text{PFD}_{\max}$ *is bounded by withheld rewards and time value.*

If external payoffs exceed this bound, endogenous incentives cannot deter attack. This is a general limitation of non-slashing designs and not unique to Crosslink Zebra.

Implication. The proposed anti-jackpot constraint (R4) eliminates endogenous ransom incentives but does not by itself deter exogenous short-seller griefing. Observability and governance response are therefore essential complements.

7 R4 as a Liveness Requirement

If pending rewards increase during stalls, the mechanism admits a rational-stall equilibrium. Preventing reward growth via cap or decay is therefore *necessary* for incentive-compatible liveness.

Cap vs. decay. A hard cap on the pending pool ($R \leq \bar{R}$) achieves the strongest property: once the cap binds, immediate finalization strictly dominates delay. Decaying per-block rewards (where each new block adds a geometrically smaller reward) bound the duration and magnitude of rational delay but do *not* make $R(\Delta)$ non-increasing—the pool still grows, just more slowly. In practice, combining both mechanisms—decaying per-block rewards with a hard cap—achieves bounded delay, a bounded pool, and eventual strict dominance of immediate finalization.

Decay semantics. If decayed rewards are burned, griefing by attackers with external short positions becomes possible. If carried forward, the hostage surplus may reappear. A third option is redirection to a community fund, which avoids both direct burn incentives and carry-forward hostage surplus but introduces governance complexity.

Safety region. Decay parameters must include a grace window exceeding expected network partitions and a rate small relative to expected annual rewards, so honest operators are not penalized for transient faults.

8 Catastrophic Liveness Failure Modes

A distinct class of failure modes arises from the interaction between the PoW chain (which continues to advance) and the Finality Gadget (which may stall and freeze the validator set). Identifying these early allows the design to incorporate mitigations before they become operational risks.

8.1 The Zombie Set (Entropic Failure)

If effective participation in the frozen set falls permanently below the finality threshold due to lost keys, bankrupt operators, or custodial policy changes, the finality gadget halts permanently.

Attrition dynamics. Even modest validator attrition compounds over time. For example, if a small percentage of active validator stake becomes unresponsive per month, a multi-month stall can reduce effective participation below threshold. Because the set is frozen, such attrition cannot be repaired endogenously. Once participation drops below threshold during a freeze, the finality gadget is permanently inoperable absent external intervention.

8.2 The Liquid Exit Paradox (Security Budget Decay)

Because validator membership and voting weights are referenced to the roster state at the last finalized block, a prolonged stall can create a mismatch between the frozen set's formal voting weight and the live economic state of stake holders.

If, during a stall, stake can fully exit its economic exposure (e.g., via unbonding/claim completing on the PoW chain) while the frozen set's voting weights remain authoritative for finality, then the cost to corrupt or bribe members of the frozen set may decline over time. In that case, the "security budget" backing finality decays, even if nominal voting weights remain constant.

Whether this risk is material depends on the staking epoch rules (locked phases, unbonding delay) and the feasibility of exit actions under stall conditions (including censorship or inclusion delays). This audit flags the issue as a conditional hazard and suggests that the specification explicitly address how staking exits interact with prolonged stalls.

Mitigation option. One possible mitigation is to constrain completion of exit (e.g., claim) during prolonged stalls, so that stake cannot fully shed exposure while its historical voting weight remains authoritative. This is a significant design choice and must be evaluated against user experience and governance expectations.

9 Degraded Mode and Liveness Recovery

After prolonged stalls, safe resumption of finality requires either:

- *Catch-up finalization*, in which the frozen validator set issues a sequence of certificates that advance finality forward from the last finalized height toward the PoW tip; or
- An explicit *liveness reset* (R5), in which governance introduces a checkpoint that finalizes the current PoW tip and establishes a new active set.

Catch-up incentives. During catch-up, accumulated rewards may be distributed across multiple updates or as a lump sum. Incentives during catch-up require care to avoid MEV extraction through selective ordering of catch-up certificates. This audit flags catch-up incentives as a topic warranting dedicated follow-up analysis.

Automatic resumption of finality directly at the PoW tip without finalizing intervening history is unsafe because validator membership and stake distribution at the tip are themselves derived from unfinalized state, creating circular dependencies between state, membership, and finality.

10 Censorship Attribution

Hybrid finality redistributes censorship power.

- Miners may censor transactions by exclusion.
- Finalizers may censor by withholding finality on blocks containing particular transactions.
- Collusion amplifies both vectors.

10.1 Targeted transaction censorship

Finalizers may withhold finality on blocks containing specific transactions. Shielded transactions obscure contents, but transparent transactions remain identifiable. This targeted censorship differs from random delay and can be used to exclude specific users or contracts. The audit does not claim this risk is eliminated, only that bounded delay incentives, observability, and governance response limit its duration and visibility.

11 Recommendations

The recommendations below are offered as spec-compatible design options. They are intended to be evaluated and refined collaboratively as the Crosslink specification evolves.

11.1 Core Recommendations

- **R1:** Pay finalizer commission only to canonical signers to prevent free-riding.
- **R2:** Pay a bounded, single-use miner bounty δR, keyed to **Finalized Height** to prevent malleability gaming.
- **R3:** Standardize telemetry for finality gap and participation.
- **R4: Anti-jackpot:** Prevent pending rewards from increasing during stalls (cap or decay) to prevent "Hostage Holdout" attacks.

11.2 Governance & Structural Extensions

- **R5: Governance Reset:** Define a governance-level mechanism to reset the active set if "Zombie Set" occurs.
- **R6: Certificate Aging:** Apply miner bounties to any certificate that advances finality, optionally weighted by blocks advanced.

- **R7 (Optional): Stall–Exit Coupling:** Explicitly specify whether and how staking exits can complete during prolonged stalls to avoid security-budget decay of the frozen set.

11.3 Operational Mitigations

- **DoS Hygiene:** Implement duplicate-signature suppression and peer-scoring to mitigate equivocation spam at the networking layer.

12 Conclusion

Crosslink Zebra's safety is structural; its liveness is economic. Rational stalls, entropic validator loss, and miner–finalizer collusion are real risks in frozen-set, non-slashing designs.

The audit identifies that "No Slashing" requires stronger compensatory controls in the networking layer (DoS protection) and strictly defined Fork Choice rules to prevent security degradation. The introduction of the **Liquid Exit** and **Malleability** vectors highlights the need for precise coupling between the consensus, networking, and economic layers.

This document is one step in an ongoing design process. As the Crosslink specification develops, the analysis here should be revisited, and the trade-offs it surfaces should inform further iteration. Feedback, critique, and refinement from the broader community are essential to that process.