Rate-Limiting Nullifiers for Gasless Sequencer Admission in Ethereum Layer-2 Rollups

Uğur Şen[1] Sergei Tikhomirov[1] Sylvain Delhomme[1] Nadeem Bhati[2] Cyprien Grau[2]

[1]Institute of Free Technology (IFT)

[2]Status Network

{ugur, sergei, sylvain, nadeem, cyp}@status.im

Abstract

Blockchain networks rely on transaction fees for resource allocation and spam prevention. Ethereum's gas mechanism and its adoption by Layer-2 rollups serve this dual purpose, but gas-based fee markets produce unintended consequences: ineffective spam deterrence at low fee levels, poor user experience, privacy leakage, and revenue instability for rollup operators. We present an idealized protocol architecture for gasless sequencer admission in Ethereum Layer-2 rollups based on Rate-Limiting Nullifiers (RLN) and a non-transferable reputation token (Karma). Users transact within a per-epoch gasless quota. Transactions beyond that quota use a gas-paid overflow path. RLN enforces the quota via zero-knowledge membership proofs, preserving pseudonymity for users within quota against on-chain observers while exposing violators through reputation slashing. We present the architecture and transaction flow, analyze spam-attack economics through a parameterized cost comparison on a flat per-identity quota model (labeled flat-N, an analysis model for the spam stress test), and describe Status Network (SN), a deployed Ethereum L2 that implements this design.

Keywords: Blockchain, Layer-2 rollups, transaction fee mechanism, rate limiting, zero-knowledge proofs, pseudonymity, MEV.

1 Introduction

Blockchains are open-access systems that require mechanisms for resource allocation. Since Bitcoin [29], transaction fees have served this purpose. Ethereum generalized the approach with gas, pricing execution proportionally to computational cost [3].

A rollup is a blockchain that executes transactions off-chain while anchoring state and transaction data to Ethereum [19]. Many rollups reuse the Ethereum-style gas mechanism. On such chains, gas measures execution cost. A fee market then prices gas to allocate block space and may also fund sequencers. Metering work, allocating block space, and earning revenue are distinct functions.

When rollup fees fall, gas still meters execution, but allocation and spam deterrence weaken. User experience (UX) suffers, privacy leaks through fee-related patterns, and operator revenue becomes unstable. Reputation-based rate limits are an alternative, but enforcing them usually links behavior to identity, which conflicts with pseudonymity in permissionless systems.

Rate-Limiting Nullifiers (RLN) [11, 33, 34] address this tension. With RLN, users prove group membership and that they satisfy the per-identity rate limit when they submit messages, without revealing their identity. Only rate-limit violators are identified, enabling reputation slashing that revokes their membership.

This paper proposes an idealized protocol architecture for gasless sequencer admission on rollups using RLN-enforced rate limits tied to a non-transferable reputation token (Karma).

Each RLN identity receives a per-epoch sequencer admission quota, addressing fee-market spam and UX failures on the within-quota path. Transactions beyond the quota use a gas-paid overflow path. RLN-based admission is enforced at the rollup sequencer. The design targets Ethereum-based rollups but applies to other environments with Merkle-tree membership contracts. Pseudonymity is graded by observer model and stated formally. We analyze the economics of sustained spam attacks under the flat-N model and describe SN [43, 42], a deployed Ethereum L2 that implements the design.

2 Background

Layer 1 (L1) denotes Ethereum mainnet, which provides consensus finality and, for rollups, settlement and data availability for batched rollup data. A rollup executes transactions on a separate chain while periodically committing state or transaction data to L1. We refer to that execution chain as *Layer 2 (L2)* relative to Ethereum.[1] The *sequencer* is the rollup's *block producer*, ordering and executing user transactions. On many rollups the sequencer is centralized, though decentralized or shared sequencing is possible [28, 7]. Our threat analysis models a centralized sequencer, except where censorship and escape hatches are discussed. A *rollup operator* is the organization that runs rollup infrastructure—typically the sequencer and often batch posting, L2 validity proving, RPC services, and verification of RLN admission proofs.

Maximal extractable value (MEV) is profit from choosing how transactions are ordered, inserted, or censored [12]. *Probing* is automated on-chain trial transactions that test whether a profitable trade is available under current or imminent state. Most probes revert. *Spam*, in this paper, is sustained automated load—including probing—that crowds out ordinary users when fees underprice block space. A *penalty* is any adverse consequence after misbehavior. *Slashing* is an on-chain penalty, where verifiable evidence of abuse triggers contract logic that updates shared state. *Financial slashing* forfeits bonded stake or collateral. *Reputation slashing* revokes membership or burns non-transferable on-chain reputation without principal forfeiture. Unless stated otherwise, slashing means reputation slashing.

2.1 Limitations of gas-based rollups

Spam, MEV, and block-space pressure

Gas-based rollups inherit Ethereum's fee market as the main gate on who gets block space [36, 9, 35]. When fees are low, spam and probing impose congestion externalities: automated transactions crowd out ordinary users, raise latency, and absorb throughput gains. Both L1 and L2 chains have seen sustained floods of such activity [45, 27]. On major rollups, MEV and probing activity are large relative to Ethereum—for example, spam has regularly consumed over 50% of gas on Base, OP Mainnet, and related OP-Stack chains, and incremental throughput on Base has been largely absorbed by spam and probing [27]. On L1, MEV extraction has concentrated block production among a small number of specialized block builders [46, 20].

Rollup sequencers often operate *private* mempools where pending transactions are sent directly to the sequencer instead of being propagated on a P2P gossip network. Searchers then fall back to probing [39]. Under first-come-first-served ordering this leads to latency races at the sequencer interface [6]. Marginal gas pricing applies the same fee to a user swap and a reverted probing transaction, so low L2 fees at typical levels often fail to deter probing-heavy load despite MEV-sharing and ordering auctions [26, 6].

[1] Layer 2 is a broader category than rollups. We use the terms *rollup* and *L2* interchangeably in this paper.

User experience (UX) and fee abstraction

Gas imposes cognitive load and an onboarding barrier, particularly for non-technical users. Users must understand gas pricing, estimate costs, and hold native tokens before transacting. That friction persists on rollups even when absolute fee levels fall.

Fee abstraction—sponsorship of gas by a third party—is the main ecosystem response. Account-abstraction paymasters (ERC-4337 [4]) enforce sponsorship rules in on-chain contract code, which is publicly auditable but still ties admission to gas accounting and bundler inclusion. Off-chain relayers run validation on private servers instead, so users trust the relayer's checks and fee quotes [38]. Both patterns keep gas in protocol accounting and chain resource metering. They shift payment and trust to a sponsor or relayer, and sponsored flows still compete for block space through the underlying fee market.

Privacy limitations

Gas-based admission adds privacy leakage through gas funding provenance and fee metadata. Users who self-fund transactions must first acquire native tokens, creating links that chain-analysis systems use as a deanonymization signal. Gas price choices and fee-payment patterns also fingerprint behavior [1, 47]. Fee abstraction (Section 2.1) hides gas payment from ordinary users on sponsored flows and mitigates UX friction and part of the gas-payment signal. Admission is still gated by the fee market, and self-funded flows keep the same provenance and fee-metadata leakage.

Operator revenue instability

Operator revenue on gas-based rollups is often tied to sequencing fees and related gas income, making it sensitive to fee compression. Rollup fees have been in a race to the bottom: the Dencun upgrade (2024-03) reduced median L2 fees by up to 99% [22, 17]. Ethereum L1 daily transaction-fee revenue has likewise fallen in the rollup era, from roughly $1–1.5 million per day through much of 2024 to a few hundred thousand dollars per day in 2026 [15]. The ecosystem response has been to supplement fee income with subsidies, ecosystem funding, and application-level monetization. These measures support operations during low-fee periods. They fund the operator off the admission path, while sequencing income still depends on volatile gas-fee conditions.

3 Design goals

Motivated by the limitations above, we adopt four design goals:

- *Spam prevention.* Block space should be allocated so that Sybil attacks are costly [14] and users receive predictable quota access even under automated load.
- *User experience.* Users should operate under an intuitive per-epoch transaction quota instead of holding native gas tokens or estimating per-transaction fees.
- *Privacy.* Honest within-quota users should remain unlinkable to their RLN identities against on-chain observers, without funding-trail or gas-price fingerprints. Only rate-limit violators should have their RLN secret exposed and be subject to slashing.
- *Revenue.* Rollup operators should be able to sustain operations without relying solely on volatile per-transaction sequencing fees.

4 Preliminaries: Rate-Limiting Nullifiers

RLN [11, 33, 34] is a cryptographic construction that enables pseudonymous, rate-limited access to a shared resource. Users register in an on-chain RLN contract and prove membership in zero knowledge without revealing their identity. Each proof is bound to an epoch. RLN guarantees that when a user exceeds their per-epoch quota, the user's secret is exposed, enabling anyone to slash them on-chain and revoke their membership.

The RLN construction consists of three functions: Register, Signal, and Slash.

Register $(sk) \to (id_{\mathsf{commit}}, i)$: Initializes a new RLN identity. The user samples a private secret sk and computes the identity commitment $id_{\mathsf{commit}} = H(sk)$, where H is a cryptographic hash function modeled as a random oracle. id_{commit} is submitted to the RLN contract. sk remains private on the client device. The RLN contract inserts the commitment into the *RLN Merkle tree* (the on-chain Merkle tree whose leaves are RLN identity commitments) and returns the corresponding leaf index i.

Signal $(sk, i, e, m, x) \to (\pi, nullifier, y)$: Generates a rate-limited signal. The inputs are the user secret sk, leaf index i, epoch e, in-quota message identifier $m \in \{1, \ldots, N\}$, and signal hash x (a hash of the transaction payload).

The signaling procedure derives $a = H(sk, e, m)$ and computes the Shamir share $y = sk + x \cdot a$ [37]. The published nullifier is $nullifier = H(a)$. The function outputs a zero-knowledge proof π attesting that $(nullifier, y)$ are derived correctly and that $H(sk) = id_{\mathsf{commit}}$ is a valid leaf of the RLN Merkle tree with the RLN contract's current root. Because sk is private and H is one-way, an external observer holding the public RLN Merkle tree cannot compute a or $nullifier$ for any leaf.

Slash $((y_1, x_1), (y_2, x_2)) \to sk$: Detects and penalizes rate-limit violations. A user who sends more than N gasless transactions in a single epoch must reuse some message identifier m, because m is restricted to $\{1, \ldots, N\}$ within an epoch. Two signals then share a and therefore the same nullifier. Given two such signals with distinct signal hashes $x_1 \neq x_2$, the Shamir shares $(y_1, x_1), (y_2, x_2)$ lie on the affine polynomial $y = sk + x \cdot a$ and allow recovery of sk by Lagrange interpolation.[2] Once a slasher submits the recovered secret to the RLN contract, the contract triggers slashing, removing the violator's RLN identity.

RLN has been studied [44] and used in decentralized messaging, notably in the Logos Messaging protocol (previously known as Waku) [34] within the Logos stack [24].

5 Rollup architecture with RLN and Karma

The flat-N rollup architecture admits users via RLN proofs tied to Karma. Each identity has a single flat per-identity quota N and enters through a stake-and-wait registration path.

5.1 Roles

The protocol involves four roles.

- *Users.* A user is any participant who submits rollup transactions. Gasless (within-quota) submissions carry an RLN proof. Obtaining RLN membership requires sufficient Karma, which accrues through staking in this model and binds to a single registrant.

[2] Identical-signal collision, where the same x is reused such that shares coincide, is scoped in Subsection 9.1.

- *Sequencer.* Beyond ordering and executing transactions (Section 2), the sequencer verifies that each incoming transaction is well-formed and accompanied by a valid RLN proof. Transactions that fail either check are dropped.
- *Slasher.* Slasher nodes monitor published blocks for duplicate nullifiers within the same epoch, which indicate rate-limit violations. When a violation is detected, the slasher recovers the violator's secret via Shamir interpolation and submits it to the contract to trigger slashing. We assume at least one honest, continuously operating slasher.
- *Contract.* The smart contract, hereafter the contract, holds user stakes and mints Karma while the stake is locked. The contract exposes the RLN Register and Slash functions. Once a user's Karma reaches the registration threshold θ, they register an RLN identity commitment on-chain. Upon a successful slash, the contract removes the violator's identity commitment from the RLN Merkle tree and applies the slashing policy.

5.2 Protocol parameters

The flat-N protocol is governed by the parameters below, each stored in or derived from the on-chain contract in the model. They are the inputs to the spam-cost analysis.

- *Stake (s).* The amount of ETH or a designated stablecoin that a user locks in the contract.
- *Threshold Karma (θ).* The minimum Karma balance required to Register one RLN identity.
- *Waiting time (Δt).* The duration for which stake s must remain locked before accumulated Karma reaches θ. Under linear Karma accrual at rate α per unit stake per unit time, $\Delta t = \theta/(\alpha \cdot s)$. Larger stake reduces Δt while still requiring lockup of s for that period.
- *Global rate limit (N).* The maximum number of gasless transactions per RLN identity per epoch. Exceeding N yields duplicate nullifiers and enables slashing (Section 4). Higher throughput requires registering multiple RLN identities, each incurring the stake-and-wait cost to reach θ.
- *Epoch length (T).* Wall-clock duration of an epoch. Each RLN proof and nullifier is valid only for one epoch (the epoch identifier e in Signal). The per-identity gasless quota N resets when the epoch changes.

5.3 Transaction flow

Each RLN identity receives a quota of N gasless *quota units* per epoch (one RLN message identifier per simple transaction in the idealized model). Beyond this quota, a user sends transactions through a gas-paid overflow path. The protocol proceeds as follows (Figure 1).

1. The user stakes ETH or a stablecoin in the contract.
2. The contract mints Karma to the user in proportion to the stake and protocol rules.
3. After waiting time Δt, once accumulated Karma reaches the registration threshold θ, the user registers an RLN identity commitment on-chain via Register.
4. For each gasless transaction, the user constructs the transaction payload, generates an RLN proof via Signal, and broadcasts the signed transaction together with the zero-knowledge proof and the *hashed RLN public bundle* to the sequencer and to slasher nodes.[3] The proof attests to RLN membership in the current epoch and a message identifier, and reveals a nullifier and Shamir share y. The hashed RLN public bundle contains only hashes derived from the RLN secret (no identity commitment or EVM signing key).

[3] If Signal is outsourced to a trusted prover, that prover learns the user–transaction linkage.

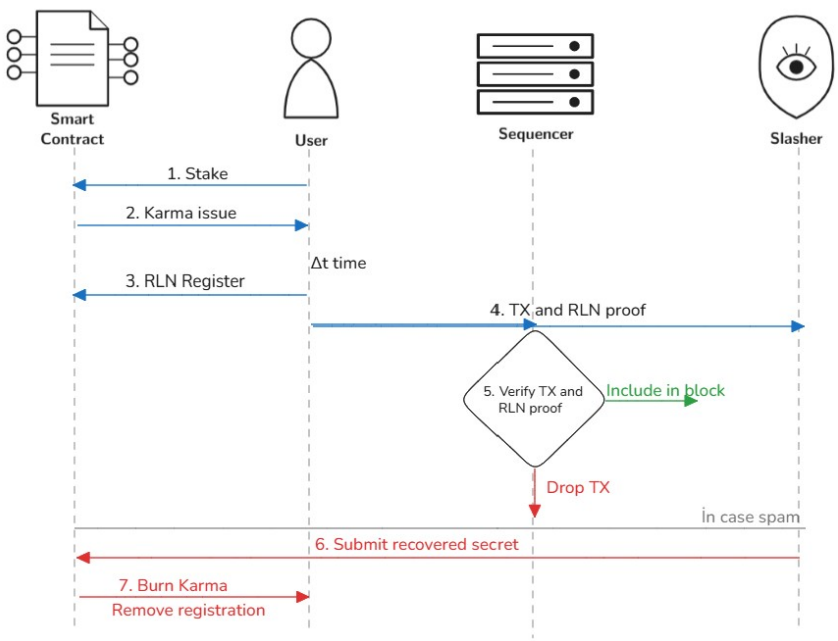

Figure 1: Protocol overview (illustrative). The numbered steps are given in the following list.

5. The sequencer verifies the RLN proof and checks that the nullifier has not appeared in the current epoch. The transaction is included in a block if and only if verification succeeds.

6. If a user exceeds the rate limit N within an epoch, duplicate nullifiers appear in published blocks. Slashing backstops cases where admission fails to keep both inclusions off-chain (dishonest sequencer or concurrent races). A slasher detects the violation, recovers the user's secret from two proofs sharing a nullifier via Slash, and submits it to the contract.

7. The contract applies reputation slashing (Section 2): it removes the violator's identity commitment from the RLN Merkle tree, burns their Karma, and mints a slasher reward according to protocol policy. Locked stake s remains withdrawable. Regaining gasless access requires re-accumulating Karma through waiting time Δt and registering a new RLN identity, so the costs are lost quota access and forgone yield during that cooldown.

5.4 Threat model and formal guarantees

We consider a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ that may control any set of protocol participants. The adversary has full visibility of the public chain state, including all published blocks, nullifiers, and stake balances, and adaptively chooses which identities to register or abandon. The proof sketches below rely on:

- *Honest verification at admission.* The sequencer checks RLN proofs and rejects duplicate nullifiers in the current epoch before including gasless transactions.
- *Slashing liveness.* At least one party that monitors published blocks submits a slash when duplicate nullifiers appear. Safety of rate enforcement depends only on public chain data. A full permissionless slasher-incentive market remains future work.
- *RLN soundness.* The RLN construction satisfies the standard soundness, completeness, and zero-knowledge properties summarized earlier (Section 4).

Rollup sequencing policy (censorship, reordering, off-protocol collusion) and prover availability are scoped separately (Subsection 9.1). Under this model, the protocol provides the following guarantees.

Theorem 1 (Gasless rate enforcement and slashing backstop)**.** *No RLN identity can have more than N gasless transactions with* distinct *signal payloads x included on-chain within a single epoch. If more than N such transactions from one identity appear in published blocks in the same epoch, that identity is slashed.*

Sketch. The sequencer rejects duplicate nullifiers in the current epoch before inclusion, so at most N distinct nullifiers from one identity can be included per epoch. Suppose nevertheless that more than N gasless transactions from one identity appear on-chain in the same epoch. Each inclusion requires a valid RLN proof bound to that epoch. By the soundness of the RLN construction [34], more than N such proofs for one identity imply reuse of a nullifier, which exposes the user's secret via Shamir interpolation. An honest slasher then submits the violation and triggers slashing. □

Observation 1 (Linear Sybil cost under stake-gated registration)**.** *An adversary that seeks k times the base gasless quota of N transactions per epoch must control at least k independent RLN identities. The only way to multiply the gasless quota is to multiply registrations, which requires locked stake s per identity for duration Δt until Karma reaches the registration threshold θ. Economic cost scales linearly in k when measured in locked stake and lock duration.*

Sketch. One identity yields at most N included gasless transactions per epoch (Theorem 1). Karma is non-transferable, so the adversary cannot consolidate k quotas on a single identity. Each registration requires the stated stake and waiting time. The resulting capital cost is bounded by the stake locked across identities and the lock duration. □

We state pseudonymity in an indistinguishability-style framework over three target levels, adopting the unlinkability terminology of Pfitzmann and Hansen [31] and the blockchain-pseudonymity framing of Bonneau et al. [2]. The levels differ in what an adversary observes besides the public chain state. They concern linkage between an RLN signal and an RLN identity. EVM senders, recipients, asset amounts, and calldata remain visible on the execution layer. Throughout, $\mathcal{K}(e)$ denotes the set of honest RLN identities eligible to signal at epoch e, and adversary-controlled registrations do not enlarge it.

Definition 1 (Level 1 pseudonymity, ideal)**.** *For any RLN identity, any epoch e, and any transaction submitted by that identity in epoch e, no PPT adversary with read access to the public chain state (blocks, nullifiers, Shamir shares, the RLN Merkle tree, and the stake registry) can identify the issuing identity with advantage non-negligibly above $1/|\mathcal{K}(e)|$. Achieving Level 1 requires a decentralized prover, since any single prover that aids signal construction necessarily learns the user–transaction linkage.*

Definition 2 (Level 2 pseudonymity, architectural)**.** *The same indistinguishability advantage as in Definition 1 holds for any PPT adversary restricted to public chain state, including the published zero-knowledge proof and the hashed RLN public bundle (Subsection 5.3), when a user within quota submits a signed transaction together with an RLN signal from* Signal*. This is the level the architecture targets for third-party observers of L2 state.*

Definition 3 (Level 3 pseudonymity, deployed)**.** *Third-party observers restricted to public chain state obtain the same indistinguishability bound as in Definition 2. Relative to a single operator-run RLN prover the Level 2 bound does not hold: the prover sees each signed gasless transaction before proving and learns which signing account maps to which RLN identity.*

We write *chain-only* for a PPT adversary with read access to public blocks, nullifiers, Shamir shares, the RLN Merkle tree, and the stake registry, but without relay timing, off-chain staking provenance, or metadata from gas-paid overflow transactions.

Theorem 2 (Pseudonymity)**.** *Modeling H as a random oracle, the architecture achieves Level 2 pseudonymity (Definition 2) against a chain-only adversary as defined above. If users rely on a single operator-run prover, they obtain Level 3 (Definition 3) relative to the rollup operator. Side-channel leakage outside this model is discussed in Subsection 5.5 and Subsection 9.1.*

Sketch. A signal observed on-chain is the tuple $(\pi, nullifier, y)$. Zero-knowledge of π contributes negligible distinguishing advantage by definition. A single Shamir share $y = sk + x \cdot a$ is uniform in the field given the unknown a and a single sample, and therefore contributes no advantage on its own. The nullifier $H(a) = H(H(sk, e, m))$ is a random-oracle output keyed by the user secret sk. Because $id_{\mathsf{commit}} = H(sk)$ is a one-way image of sk and the hashed RLN public bundle does not reveal sk, no PPT adversary with read access to chain state can compute candidate nullifiers per leaf, so the posterior over $\mathcal{K}(e)$ is statistically close to uniform (Level 2). If the rollup operator's RLN prover receives the signed transaction before proving, it learns the mapping from signing account to RLN identity, which is the Level 3 guarantee. Third parties still observe only the hashed RLN public bundle on-chain. □

5.5 Side channels and further adversaries

The following side channels fall outside the guarantees above (Theorem 1, Theorem 2, and Observation 1).

The pseudonymity bounds ignore information leakage outside the indistinguishability game. On-chain, stake balances and registration timestamps enable staking-pattern fingerprinting. Overflow transactions paid in gas reintroduce funding-trail and gas-price fingerprints. Execution payloads still reveal accounts, recipients, amounts, and calldata. Off-chain, gossip-network observers correlate broadcast timing and source addresses. Closing those channels is outside RLN admission and needs complementary mechanisms at other layers (e.g., mix networks for broadcast metadata, or application-layer shielded flows for execution payloads), each under its own trust assumptions.

The waiting time Δt mitigates flash-loan-funded registration in the flat-N model [32]. Over-the-counter purchase of dormant RLN identities is weakened by non-transferable Karma and low balances on inactive accounts.

5.6 Extensions

Epoch boundaries Quota is enforced per epoch. Sliding wall-clock windows are handled separately below. The two diverge at epoch boundaries as follows. If epochs are synchronized across users, an identity consumes up to N gasless slots at the end of one epoch and up to N at the start of the next, so a sliding window of length T aligned with epoch length admits on the order of $2N$ transactions (standard boundary-doubling under fixed epochs). This paper does not prove a tighter sliding-window bound. Alternative epoch and quota designs soften bursts.

We nevertheless use fixed epochs because they keep nullifier scope and contract state keyed to a single epoch index, matching how RLN proofs bind to e (Section 4). Sliding-window enforcement would require rolling nullifier domains or heavier on-chain bookkeeping.

Optional Karma-based intra-block ordering RLN admission decides which transactions execute gaslessly within quota. Congestion beyond quota is handled by the gas-paid overflow path. Intra-block ordering among admitted transactions is a separate, *optional* extension beyond inclusion-only sequencer policy. When demand exceeds block capacity, some ordering rule is unavoidable. On gas-based rollups with private mempools (Section 2.1), congestion often becomes a gas-price or latency-race contest at the sequencer [6]. After proofs and nullifiers are validated, the sequencer sorts admitted gasless transactions in non-increasing order of Karma-derived score $\rho(K)$, where K is the Karma balance at block construction, with ties broken from the hashed

Table 1: Spam-economics notation (analysis inputs and derived quantities).

Symbol	*Meaning*
R	Target sustained attack rate (transactions per second) in the stress test.
k	Adversary-controlled RLN identities, at least $\lceil RT/N \rceil$ at rate R over epoch T.
r	Annualized yield on locked stake used in C_{attack}.
S_{locked}	Total stake locked across identities (ks).
C_{attack}	Forgone yield on S_{locked} over lock duration $\Delta t + T$.
T_{attack}	Gasless transactions in one attack epoch (RT).
T_{year}	Seconds per year (31,536,000), normalizes r in C_{attack}.

RLN public bundle (e.g., lexicographic nullifier order). Gas-paid overflow keeps the sequencer's existing gas ordering. We sketch the policy without prescribing a concrete ρ.

Under load, ordering by $\rho(K)$ rewards users with more accumulated Karma without reopening a per-send fee auction on every gasless transaction. Ordering by $\rho(K)$ complements sequencing proposals (e.g., Timeboost [6]) that target mempool privacy and auction design, while Karma ordering targets fairness among senders who already passed stake-gated admission.

A user who splits stake across m identities with about K/m Karma each gains multiple moderate block positions unless ρ is *subadditive*: $\sum_{i=1}^{m} \rho(K/m) \leq \rho(K)$ for $K > 0$ and integers $m \geq 2$ (linear $\rho(K) = K$ sits on the boundary: the inequality holds with equality, so splitting neither helps nor hurts; faster-than-linear growth worsens splitting).

Public ordering by $\rho(K)$ leaks coarse reputation rank beyond the pseudonymity bound (Theorem 2). *Bucketed* prioritization (priority class only, pseudorandom order within class keyed by block hash and nullifier) is one possible mitigation. Per-identity quota N and RLN registration stay unchanged under this ordering policy.

Karma and protocol governance Governance design is orthogonal to the core RLN admission rules, but it shapes long-run incentives around staking and participation. An optional extension grants Karma holders governance rights with voting weight tied to Karma balance, encouraging users to keep stake locked after obtaining RLN membership. Ballots, timelocks, quorum rules, and the interaction between governance and slashing are deployment policy choices outside the core RLN admission rules.

6 Economics of spam deterrence

The flat-N analysis quantifies spam deterrence through locked stake $S_{\mathsf{locked}} = ks$ and opportunity cost C_{attack} at yield r. Notation is summarized in Table 1. Protocol constants s, θ, Δt, N, and T are defined in Subsection 5.2. The numerical stress test is an analysis instrument for this model.

The binding constraint for a capital-limited adversary is throughput. At aggregate locked stake $S_{\max}$, feasible rate satisfies $R \leq S_{\max} N/(sT)$ because each identity locks stake s and sends at most N gasless transactions per epoch of length T (formalized below). Opportunity cost on that locked principal is a separate flow metric.

6.1 Stress test: parameters and gas–RLN comparison

The stress test uses $\theta = 3$ Karma, $s = 0.13\,\text{ETH}$ per identity, $\Delta t = 86{,}400\,\text{s}$ (one day), $N = 1$ gasless transaction per epoch, $T = 3{,}600\,\text{s}$ (one hour), $R = 100\,\text{TPS}$, and $r = 0.03$ (3% annual yield). The scenario is illustrative for the flat-N model. Rationale for each value is given below.

- *Epoch length ($T = 3{,}600\,s$).* One hour matches the attack horizon used when comparing liquid gas spend to RLN opportunity cost at $R = 100\,\text{TPS}$, so both sides of the comparison refer to the same wall-clock window. Shorter epochs tighten nullifier scope and reset quotas

more often, increasing proof and contract churn. Longer epochs slow how quickly rate limits respond to abuse.

- *Per-epoch quota ($N = 1$).* With $N = 1$, sustaining rate R for duration T requires the maximum identity count $k = \lceil RT/N \rceil$ implied by Observation 1, making the spam-cost calculation conservative relative to settings where $N > 1$.
- *Waiting time ($\Delta t = 86{,}400\,s$).* One day exceeds typical flash-loan settlement latency (seconds to minutes), so registration cannot be completed inside a single atomic loan. We treat θ, s, and Δt as independent inputs so the opportunity-cost formula stays explicit (the linear-accrual identity $\Delta t = \theta/(\alpha s)$ would otherwise tie Δt to minting rate α).
- *Threshold Karma ($\theta = 3$).* A small integer threshold keeps the registration gate easy to read in formulas and sensitivity plots. Karma is counted in abstract units in the model.
- *Stake ($s = 0.13\,ETH$).* Stake per identity scales both locked capital $S_{\mathsf{locked}} = ks$ and opportunity cost C_{attack} below. The value is chosen as an order-of-magnitude illustration compatible with the $k = 360{,}000$ identity count at $(R, T) = (100\,TPS, 1\text{ hour})$ and $N = 1$.

Gas-based versus RLN probing at fixed (R, T) Suppose an adversary seeks sustained on-chain probing at rate $R = 100$ transactions per second for one hour ($T = 3{,}600$ s, i.e., 360,000 transactions), compatible in order of magnitude with bot-dominated load on high-throughput L2s [27]. On a gas-based rollup, public fee trackers report swap-class transactions at roughly \$0.08–\$0.10 on Optimism [23, 39]. We use $p = \$0.08$ per transaction conservatively. Liquid fee spend for the hour is $C_{\mathsf{gas}} \approx 360{,}000 \times p \approx \$28{,}800$. If the adversary's *liquid* budget per epoch is $B = C_{\mathsf{gas}}$, the fee-limited probing rate is $X = B/(pT) \approx 100\,\text{TPS}$. The same (R, T) is analyzed under the RLN parameters stated above.

For MEV-style probing, opportunity cost on staked capital is the main flow metric in the RLN cost model. At the stress-test parameters it is about 4.01 ETH (approximately \$10,017 at \$2,500/ETH), which is *lower* than C_{gas} at the same R, but deterrence also relies on the full notional $S_{\mathsf{locked}} \approx 46{,}800\,\text{ETH}$ to field $k = \lceil RT/N \rceil$ identities, the $\Delta t = 1$ day wait before each new identity registers, and principal lockup across k accounts.

Under a separate *capital* constraint, feasible RLN throughput falls far below the gas fee-limited rate X. An adversary who locks at most $S_{\max}$ ETH in aggregate satisfies $k \le S_{\max}/s$ and thus $R \le S_{\max} N/(sT)$. For example, $S_{\max} = 100\,\text{ETH}$ gives $R \lesssim 0.2\,\text{TPS}$, versus $X \approx 100\,\text{TPS}$ for the same liquid fee budget on gas. Pre-positioning many identities over Δt raises sustained R over multi-day horizons. Within a single hour, R stays capped by available stake and k.

We therefore separate a *fee-vs.-yield cost ratio at fixed R* (opportunity cost alone falls to about 35% of liquid gas spend when principal is unbounded) from *throughput limits* imposed by quotas, waiting time, and stake. Under this model the capital constraint is the main spam deterrent.

An adversary sustaining rate R over one epoch of length T must send $T_{\mathsf{attack}} = RT$ gasless transactions and therefore control at least $k \ge \lceil RT/N \rceil$ independent RLN identities. Each identity requires stake s held through waiting time Δt until Karma reaches θ.

Opportunity cost is the forgone yield on locked stake. During the attack epoch of duration T, stake remains locked, so the total lock duration per identity is $\Delta t + T$. For an adversary staking s per identity across k identities for a duration $\Delta t + T$, the total opportunity cost is

$$C_{\mathsf{attack}} = r \cdot s \cdot k \cdot \frac{\Delta t + T}{T_{\mathsf{year}}},$$

where $T_{\mathsf{year}} = 365 \times 24 \times 3600 = 31{,}536{,}000$ seconds is the number of seconds in a year. The adversary additionally incurs the operational burden of provisioning, registering, funding, and managing k independent identities. Those costs are omitted from C_{attack}.

Numerical example

Using the stress-test parameters above, we compute the resources required for an adversary to sustain $R = 100$ TPS over one epoch. The total transactions required are $T_{\mathsf{attack}} = R \cdot T = 100 \times 3{,}600 = 360{,}000$. Since each identity provides $N = 1$ transaction per epoch, $k = 360{,}000$ identities are required. Each identity requires $s = 0.13$ ETH staked, so the total capital locked is $S_{\mathsf{locked}} = 0.13 \times 360{,}000 = 46{,}800$ ETH. At an ETH price of \$2,500, this is \$117 million in upfront capital that must be sourced and locked before the attack uses the gasless lane. The opportunity-cost calculation below isolates the flow cost of that locked capital. Each identity must be staked for $\Delta t = 1$ day before registration and remains locked during the attack epoch $T = 1$ hour, for a total lock duration $\Delta t + T = 90{,}000$ s. At $r = 3\%$ annualized yield, the opportunity cost of locking 46,800 ETH for that duration is

$$C_{\mathsf{attack}} = 0.03 \times 46{,}800 \times \frac{90{,}000}{31{,}536{,}000} \approx 4.01\, ETH.$$

At an ETH price of \$2,500, this corresponds to approximately \$10,017, or about 35% of C_{gas} at the same (R, T).

Beyond the opportunity-cost flow, the comparison has three further asymmetries. First, the attacker must command principal on the order of \$117 million up front. After slashing, identities must be re-provisioned and wait through Δt again, while stake principal remains withdrawable and a gas-based spammer faces no comparable cooldown. Even temporary loss of gasless access disrupts latency-sensitive probing. Second, gas-based fee markets are volatile, with fees spiking under congestion and falling toward near-zero when activity is low. Under the labeled analysis assumptions, RLN ties deterrence to capital lockup and identity count at the target rate R, largely independent of short-term fee swings. Third, provisioning and managing 360,000 independent identities is a practical barrier that gas-based single-wallet spam avoids (Observation 1).

Sensitivity analysis

Attack opportunity cost scales linearly with stake per identity (s), waiting time (Δt), and yield rate (r), and inversely with the transaction quota per identity (N), when each parameter is varied while the others stay at the stress-test baseline (Figure 2). Epoch duration (T) scales cost roughly with $k = \lceil RT/N \rceil$ at fixed target rate R, so longer epochs raise both identity count and lock duration $\Delta t + T$. The Karma threshold (θ) determines eligibility only. Once identities are registered, attack cost is driven by stake lock and opportunity cost. The threshold couples with Δt and s through $\Delta t = \theta/(\alpha s)$ when Karma accrues linearly at rate α. Increasing Δt to approximately 2.9 days (2.9× the baseline) or increasing s to approximately 0.37 ETH (2.9× the baseline) would bring the RLN attack cost to parity with the gas baseline in the sensitivity plot (Figure 2).

Limitations of the linear cost model

The linear model may understate long-horizon pre-provisioning. An adversary who spreads registration over many days amortizes Δt across time. Non-linear accrual with idle decay further discourages large dormant identity inventories.

7 Status Network and simulation

SN implements the architecture as a deployed Ethereum L2. A standalone simulation measures client-side RLN proving and verification cost.

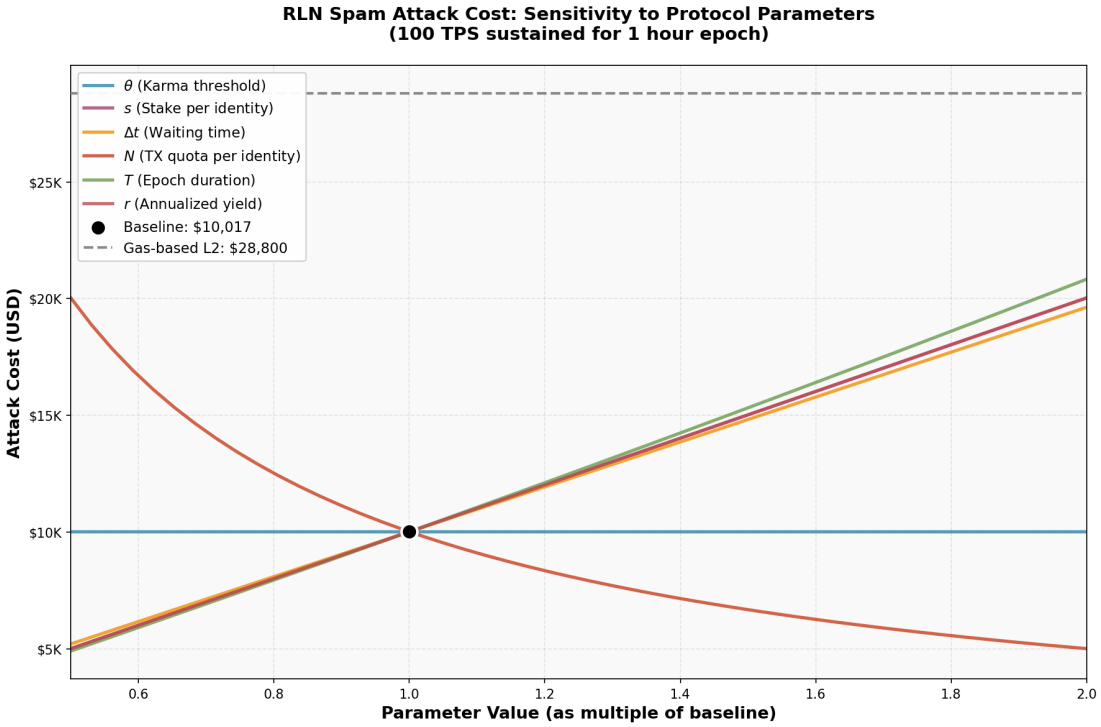

Figure 2: Sensitivity of spam attack cost to protocol parameters. Each colored line shows how the opportunity cost of sustaining 100 TPS for one epoch varies as a single parameter changes (expressed as a multiple of its baseline value), holding all others fixed. The baseline configuration (black dot at 1.0×) yields an attack cost of $10,017, about 35% of the cost of an equivalent attack on a gas-based L2 (horizontal dashed line at $28,800).

7.1 Status Network

SN [43, 42] implements the architecture above as an independent Ethereum Layer-2 rollup built on a zkEVM-style execution stack [10]. Product documentation covers gasless admission and Karma [40, 41].

Economics Rollup operations are funded from yields on bridged L1 assets deployed in standard DeFi strategies—for example, liquid-staking receipts on ETH and lending-market placements for stablecoins—together with a share of application fees on the L2 [42].

Karma on SN is non-transferable [13] and is earned through protocol-defined activities (including SNT staking and other distributors), rather than solely through the architecture's ETH/stablecoin stake-and-wait path [41]. Positive Karma unlocks gasless access. Reputation bands map to distinct per-epoch gasless quotas (tiers) [41, 40]. RLN membership registration does not require a separate stake deposit beyond the Karma gate.

SN today uses global epochs on the order of a day [40].[4]

Centralized prover and privacy SN's operator prover pipeline registers eligible addresses into the RLN Merkle tree, generates Groth16 membership proofs for gasless transactions [16], and streams proofs to a sequencer-side verifier that matches them to pending transactions [40, 43]. The rollup operator therefore receives each signed gasless transaction before proving and forwards

[4]Future hardening could include sublinear Karma accrual with idle decay, mint-weight splits, and tier-dependent epochs. Illustrative forms: $\mathsf{karma}(t, w) = \min(\theta, \kappa \cdot t^{\beta} \cdot w^{\gamma})$ with $\beta, \gamma < 1$; idle decay $\mathsf{karma} \leftarrow \max(0, \mathsf{karma} - \delta \cdot (t - t_{\mathsf{last}}))$; example mint-weight splits such as liquidity/application usage (60%), native-token staking (35%), and sequencer tips (5%).

the proved transaction with its hashed RLN public bundle to the sequencer. Users receive Level 3 pseudonymity relative to the rollup operator (Definition 3). Decentralizing RLN proving is planned future work.

Gas-weighted admission, deny list, and slashing RLN bounds admitted signals per identity per epoch. SN meters estimated L2 gas so that larger transactions consume multiple quota units before gasless admission [40]. Heavy calls consume more of the epoch budget than simple transfers. The spam-cost model in Section 6 counts one identifier per attack transaction, which is conservative for the attacker when attack transactions are gas-heavy. If the budget is exhausted, or after quota abuse, the user may be placed on an epoch-scoped deny list, use a gas-paid overflow path (Status documentation: premium gas) that bypasses RLN and restores gasless budget, or wait for the next epoch (up to one epoch) [40]. Quota exhaustion and abuse therefore interact with the deny list and the gas-paid overflow path.

On slash, SN removes the identity from the RLN Merkle tree and burns Karma, with stake principal remaining withdrawable. A configurable share of slashed Karma may reward the reporter; the remainder is burned [40, 43]. Production percentages are left to deployment configuration.

Slash submissions are made by the operator, by automated slasher services, or by both, with commit–reveal available to reduce front-running among reporters.

7.2 Simulation microbenchmark

We implemented a standalone simulation of RLN-based transaction submission in Rust [48], modeling the architectural client-side submission path (SN uses a centralized prover instead). The user attaches a zero-knowledge proof and the hashed RLN public bundle to each transaction, so the mocked sequencer cannot link the signing account to an RLN leaf from the hashed RLN public bundle alone. The simulation exercises client-side proof generation and delivery to a mocked sequencer and slasher under two scenarios: valid transactions admitted to the mempool and transactions from a user exceeding the RLN rate limit. On a consumer-grade machine, proof generation on the user (prover) side took 340 ms on average, and proof verification at the sequencer side took 2.47 ms (mean over three benchmark runs). Proofs were 288 bytes. Verification cost is small relative to generation under this single-machine configuration. The measurements are a functionality check and a microbenchmark of proving and verification cost.[5]

8 Related work

RepChain [21] maintains a reputation chain alongside transaction processing on a sharded blockchain. Other work uses reputation to reward verifiable contributions in distributed networks [25]. STARVESPAM [8] scores peers and adaptively rate-limits transactions at the Ethereum transaction relay layer. During the Otherside NFT spam event it reportedly blocked over 95% of spam while dropping about 3% of honest transactions [30, 5, 8].

Flashbots' MEV-Share [26] lets users share selected pending transactions with searchers, who bid to back-run them (place a transaction immediately after the target to profit from the state change it creates). Matched bundles (groups of transactions executed together atomically) run inside trusted execution environments, improving privacy for those bundles while routine open-mempool admission stays on the fee market. Arbitrum's Timeboost [6] auctions short-lived ordering priority via sealed bids while gas remains the primary admission market. OP-Stack teams have also proposed transaction-filtering ideas [27].

[5] Reproducibility materials accompany the simulation implementation [48]. Environment and toolchain: Ubuntu 25.10, Rust 1.94.1, and RLN parameters (Groth16 proof system [16], RLN Merkle tree depth 20, arkworks 0.5.0).

Table 2: Feature comparison.

Approach	*Gasless*	*Sybil stake*	Pseudo-nymity	*Rep. slash*	*Fin. slash*	*Seq. adm.*
ERC-4337 paymasters [4]	partial	no	partial	no	no	partial
Off-chain relayers [38, 18]	partial	no	partial	no	no	no
MEV-Share [26]	no	no	partial	no	no	no
Timeboost [6]	no	no	partial	no	no	partial
RepChain [21]	no	partial	no	partial	no	no
STARVESPAM [8]	no	partial	partial	no	no	no
This work	yes	yes	yes	yes	no	yes

Column definitions.

Gasless: within-quota execution without per-send gas from the user (overflow still uses gas). *Sybil stake*: bonded stake and/or non-transferable reputation gates at registration (SN: Karma earn paths; Section 7). *Pseudonymity*: within-quota RLN signals unlinkable from on-chain identity for third-party chain observers (Level 2; operator prover is Level 3). *Rep. slash*: reputation slashing (Section 2); *yes* only for proof-triggered updates at rollup admission, *partial* for other on-chain reputation penalties, *no* for relay-local or off-chain only. *Fin. slash*: forfeiture of bonded stake or collateral on verified abuse. *Seq. adm.*: enforcement at rollup sequencer admission before execution.

Table 3: Novelty of our architecture relative to each baseline.

Baseline	*Adds relative to baseline*
ERC-4337 paymasters [4]	Stake-backed quotas, RLN proofs, and reputation slashing at admission.
Off-chain relayers [38, 18]	Sequencer rate limits and reputation slashing with on-chain admission checks.
MEV-Share [26]	Per-epoch admission quotas enforced before execution.
Timeboost [6]	RLN-enforced admission quotas as the primary gasless gate.
RepChain [21]	Proof-triggered reputation slashing and sequencer-gated RLN on a general-purpose rollup.
STARVESPAM [8]	Sequencer admission and reputation slashing at the rollup execution gate.

Fee abstraction through ERC-4337 paymasters and off-chain relayers is summarized earlier (Section 2.1). Those patterns sponsor or relay gas while admission at the rollup sequencer remains gas-based. Paymasters [4] still pay gas on-chain and tie rate limits to account identity, so within-quota sends remain linkable to that account relative to the Level 2 target in Definition 2. Off-chain relayers [38, 18] see user intent and funding paths, with admission control at the relayer rather than at the rollup sequencer.

No surveyed mechanism combines gasless quota, Sybil-gated registration, within-quota pseudonymity, proof-triggered reputation slashing, and sequencer admission (Table 2 and Table 3).

9 Limitations and future work

The flat-N claims rest on modeling assumptions. Several practical gaps suggest directions for further research.

9.1 Limitations

The threat model assumes a centralized sequencer trusted for liveness and ordering, as on most current rollups (including SN). Escape hatches, forced inclusion on L1, and full censorship resistance are deployment and L1 policy choices. Proof sketches assume honest sequencer verification of proofs and nullifiers: a malicious sequencer can collude with users to bypass rate limits off-protocol or selectively censor valid RLN proofs, and slashing backstops only violations that still appear on-chain (Subsection 5.3). Slashing burns non-transferable Karma and revokes gasless access but leaves stake withdrawable, so a one-shot adversary may treat Karma reaccrual as a delay rather than a capital loss. Slashing liveness is assumed via an operator or automated monitor path in SN (Section 7). Spam analysis uses capital cost (Section 6), not classical transaction-fee-mechanism incentive compatibility or equilibria of bids or probing strategies (including Stackelberg or Nash models that combine MEV rents with C_{attack}). The stress test leads with capital-bounded throughput ($S_{\max}$, k, s, T, N), while opportunity cost C_{attack} is a separate flow metric sensitive to yield when principal is unbounded. Gasless admission is metered in per-epoch quota units per RLN identity (SN's gas-weighted units partially align debits with estimated execution gas [40]), rather than multidimensional fee schedules over compute, storage, and bandwidth [36, 9]. Sybil cost is linear in identities under stake-and-wait registration (Observation 1). SN adds qualitative Sybil friction via Karma earn paths and tiers [41]. Evaluation covers proving and verification microbenchmarks (Subsection 7.2); sequencer-side verification is cheap in that simulation. Chain-only pseudonymity (Theorem 2) excludes staking-pattern, gas-paid overflow, optional Karma ordering, and gossip-layer leakage (Subsection 5.5, Section 5.6). With globally aligned epochs, a sliding window of length T can admit on the order of $2N$ transactions per identity (Section 5.6). Gasless rate enforcement assumes distinct signals x (Theorem 1); reusing the same x so that Shamir shares coincide may evade secret recovery. The rollup operator's centralized RLN prover yields Level 3 pseudonymity (Definition 3): it sees each signed gasless transaction before proving and learns the signing-account to RLN-identity map. Karma-ranked intra-block ordering is an optional extension only (Section 5.6).

9.2 Future work

Directions for further research include:

- *Decentralized RLN proving.* Client-side or multi-party proving so the Level 2 bound also holds against the operator (Definition 2, Definition 3).
- *Signal binding.* Protocol changes preventing signal collision from blocking secret recovery.
- *Sliding-window quotas.* Rate limits that hold when epochs are per-identity or randomly aligned, beyond global epoch boundary doubling.
- *Side-channel privacy.* Indistinguishability bounds under staking, overflow, ordering, and gossip adversaries in established anonymity frameworks.
- *Intra-block ordering.* Formal and empirical study of Karma scores ρ, including subadditivity against identity-splitting and bucketed, randomized within-tier order to limit rank leakage.
- *Engineering evaluation.* Parameter sweeps, malformed-proof admission under concurrent load, and the operational cost of large identity sets under sustained spam.
- *Game theory and Sybil cost.* Equilibrium probing models relative to the gas-market stress test (Section 6.1), and closed-form Sybil-cost lower bounds beyond linear stake accounting.
- *Slasher markets.* Permissionless incentives for monitoring and reporting when rewards are non-transferable Karma (or a configurable reporter share of slashed Karma).

10 Conclusion

We presented an idealized architecture for within-quota gasless sequencer admission on Ethereum rollups, using RLN-enforced quotas and slashing in place of marginal gas pricing on that path. RLN preserves pseudonymity for within-quota users against on-chain observers while exposing rate-limit violators through reputation slashing. Status Network implements the design as a deployed L2, funding operations through bridged-asset yields and application fees, with operator proving, tiers, and a gas-paid overflow path documented separately. Under the flat-N model, sustained spam requires locked capital that scales linearly with attack rate, while honest users obtain predictable gasless capacity through non-transferable reputation.

References

[1] Ferenc Beres, Istvan A. Seres, Andras A. Benczur, and Mikerah Quintyne-Collins. Blockchain is watching you: Profiling and deanonymizing Ethereum users. In *2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pages 69–78, Los Alamitos, CA, USA, August 2021. IEEE Computer Society. doi:10.1109/DAPPS52256.2021.00013.

[2] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121. IEEE, 2015. URL: https://ieeexplore.ieee.org/document/7163021/, doi:10.1109/SP.2015.14.

[3] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform, 2013. Accessed: 2026-03-13. URL: https://ethereum.org/whitepaper/.

[4] Vitalik Buterin, Yoav Weiss, Dror Tirosh, Shahaf Nacson, Alex Forshtat, Kristof Gazso, and Tjaden Hess. ERC-4337: Account abstraction using alt Mempool. Ethereum Improvement Proposals, 2021. Accessed: 2026-03-13. URL: https://eips.ethereum.org/EIPS/eip-4337.

[5] Mingjia Cao, Qianchen Li, Huashan Zhou, Zhe Wang, Qian Zhu, Minhui Zhang, Tao Xie, and Yuqing Zhang. EVM*path: Dissecting the dark forest of NFT drops. In *Proceedings of the 44th IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023. Includes analysis and case studies of large-scale NFT drops and related transaction dynamics on Ethereum, including high-profile land sale events like Otherside. URL: https://www.computer.org/csdl/proceedings-article/sp/2023/933600a584/1IiFYLWVB5u.

[6] Agostino Capponi and Brian Zhu. Auctioning time to mitigate latency races: Theory and evidence from blockchains, 2026. URL: https://arxiv.org/abs/2512.10094, arXiv:2512.10094.

[7] Margarita Capretto, Martín Ceresa, Antonio Fernández Anta, Pedro Moreno-Sánchez, and César Sánchez. A secure sequencer and data availability committee for rollups. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2129–2143. ACM, 2025. URL: https://arxiv.org/abs/2509.06614, doi:10.1145/3719027.3765187.

[8] Rowdy Chotkan, Bulat Nasrulin, Jérémie Decouchant, and Johan Pouwelse. STARVESPAM: Mitigating spam with local reputation in permissionless blockchains. In *2025 7th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 1–9, 2025. doi:10.1109/BRAINS67003.2025.11302925.

[9] Hao Chung and Elaine Shi. Foundations of transaction fee mechanism design. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023. URL: https://dblp.org/rec/conf/soda/ChungS23.html.

[10] ConsenSys. Linea documentation, 2026. Accessed: 2026-03-13. URL: https://docs.linea.build/.

[11] RLN Contributors. Rate-limiting nullifier (RLN) documentation, 2026. Accessed: 2026-03-13. URL: https://rate-limiting-nullifier.github.io/rln-docs/.

[12] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020. URL: https://arxiv.org/abs/1904.05234, doi:10.1109/SP40000.2020.00040.

[13] Tim Daubenschütz and Anders. ERC-5192: Minimal soulbound NFTs. Ethereum Improvement Proposals, July 2022. Accessed: 2026-03-13. URL: https://eips.ethereum.org/EIPS/eip-5192.

[14] John R. Douceur. The Sybil attack. In *Peer-to-Peer Systems (IPTPS)*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002. doi:10.1007/3-540-45748-8_24.

[15] Etherscan. Ethereum daily transaction fee chart, 2026. Accessed: 2026-08-04. URL: https://etherscan.io/chart/transactionfee.

[16] Jens Groth. On the size of pairing-based non-interactive arguments. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 305–326. Springer, 2016.

[17] growthepie. Chain revenue, 2025. Accessed: 2026-08-04. URL: https://www.growthepie.com/fundamentals/fees-paid-by-users.

[18] Ethereum gas station network (GSN). OpenGSN Documentation, 2024. Accessed: 2026-03-13. URL: https://docs.opengsn.org/.

[19] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. SoK: Layer-two blockchain protocols. In *Financial Cryptography and Data Security*, volume 12059 of *Lecture Notes in Computer Science*, pages 201–226. Springer, 2020. URL: https://eprint.iacr.org/2019/360, doi:10.1007/978-3-030-51280-4_12.

[20] Lioba Heimbach, Lucianna Kiffer, Christof Ferreira Torres, and Roger Wattenhofer. Ethereum's proposer-builder separation: Promises and realities. *arXiv preprint arXiv:2305.19037*, 2023. URL: https://arxiv.org/abs/2305.19037.

[21] Chenyu Huang, Zeyu Wang, Huangxun Chen, Qiwei Hu, Qian Zhang, Wei Wang, and Xia Guan. RepChain: A reputation-based secure, fast, and high incentive blockchain system via sharding. *IEEE Internet of Things Journal*, 8(6):4291–4304, 2021. doi:10.1109/JIOT.2020.3028449.

[22] L2BEAT. Onchain costs, 2026. Accessed: 2026-03-13. URL: https://l2beat.com/scaling/costs.

[23] L2Fees. Layer-2 transaction fees, 2025. Accessed: 2026-03-13. URL: https://l2fees.info/.

[24] Logos. Logos documentation, 2026. Accessed: 2026-03-13. URL: https://github.com/logos-co/logos-docs.

[25] Elnaz Mehraein, Maqsood Ahamed Abdul Careem, and Aveek Dutta. "when reputation is currency": Scalable Blockchain to enforce spectrum policies. *IEEE Transactions on Cognitive Communications and Networking*, 12:688–702, 2026. doi:10.1109/TCCN.2025.3564590.

[26] Robert Miller. MEV-share: Programmably private orderflow to share MEV with users. Flashbots Collective Forum, 2023. Accessed: 2026-03-13. URL: https://collective.flashbots.net/t/mev-share-programmably-private-orderflow-to-share-mev-with-users/1264.

[27] Robert Miller. MEV and the limits of scaling. Flashbots Writings, June 2025. Accessed: 2026-03-13. URL: https://writings.flashbots.net/mev-and-the-limits-of-scaling.

[28] Shashank Motepalli, Luciano Freitas, and Benjamin Livshits. SoK: Decentralized sequencers for rollups, 2023. URL: https://arxiv.org/abs/2310.03616, arXiv:2310.03616.

[29] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. Accessed: 2026-08-04. URL: https://bitcoin.org/bitcoin.pdf.

[30] NFT Now. Otherside: Everything to know about the BAYC's metaverse, 2023. Accessed: 2026-03-13. URL: https://nftnow.com/guides/otherside-everything-you-need-to-know/.

[31] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. Technical Report v0.34, TU Dresden, August 2010. Accessed: 2026-05-26. URL: https://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.

[32] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the DeFi ecosystem with flash loans for fun and profit. In *Financial Cryptography and Data Security*, volume 12674 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2021. URL: https://arxiv.org/abs/2003.03810, doi:10.1007/978-3-662-64322-8_1.

[33] Logos Research. Rate-limiting nullifier (RLN), 2024. Accessed: 2026-03-13. URL: https://research.logos.co/rln.

[34] Alvaro Revuelta, Sergei Tikhomirov, Aaryamann Challani, Hanno Cornelius, and Simon Pierre Vivier. Message latency in Waku relay with rate-limiting nullifiers. *Cryptology ePrint Archive, Paper 2024/1073*, 2024. URL: https://eprint.iacr.org/2024/1073.

[35] Tim Roughgarden. Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559, 2020. URL: https://arxiv.org/abs/2012.00854, arXiv:2012.00854.

[36] Tim Roughgarden. Transaction fee mechanism design, 2023. URL: https://arxiv.org/abs/2106.01340, arXiv:2106.01340.

[37] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979. doi:10.1145/359168.359176.

[38] Karandeep Singh. Why we built a Paymaster contract instead of a Relayer service. Shinobi Cash, 2025. Accessed: 2026-03-13. URL: https://hackmd.io/@karandeepsingh/rylyWJcOWl.

[39] Ozan Solmaz, Lioba Heimbach, Yann Vonlanthen, and Roger Wattenhofer. Optimistic MEV in Ethereum layer-2s: Why blockspace is always in demand. *arXiv preprint arXiv:2506.14768*, 2025. URL: https://arxiv.org/abs/2506.14768.

[40] Status Network. Gasless transactions: RLN and reputation-based access, 2026. Accessed: 2026-07-21. URL: https://docs.status.network/overview/general-info/gasless-transactions.

[41] Status Network. Karmic tokenomics, 2026. Accessed: 2026-07-21. URL: https://docs.status.network/overview/tokenomics/karmic-tokenomics.

[42] Status Network. Status network documentation, 2026. Accessed: 2026-07-21. URL: https://docs.status.network/.

[43] Status Network. Status network monorepo, 2026. Accessed: 2026-08-04. URL: https://github.com/Linea-Consortium/status-network-monorepo.

[44] Sanaz Taheri-Boshrooyeh, Oskar Thorén, Barry Whitehat, Wei Jie Koh, Onur Kilic, and Kobi Gurkan. WAKU-RLN-RELAY: Privacy-preserving peer-to-peer economic spam protection. In *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 73–78, 2022. doi:10.1109/ICDCSW56584.2022.00022.

[45] Christof Ferreira Torres, Albin Mamuti, Ben Weintraub, Cristina Nita-Rotaru, and Shweta Shinde. Rolling in the shadows: Analyzing the extraction of MEV across layer-2 rollups. *arXiv preprint arXiv:2405.00138*, 2024. URL: https://arxiv.org/abs/2405.00138.

[46] Anton Wahrstätter, Liyi Zhou, Kaihua Qin, Davor Svetinovic, and Arthur Gervais. Time to bribe: Measuring block construction market. *arXiv preprint arXiv:2305.16468*, 2023. URL: https://arxiv.org/abs/2305.16468.

[47] Mike Wu, Will McTighe, Kaili Wang, István András Seres, Nick Bax, et al. Tutela: An open-source tool for assessing user-privacy on Ethereum and Tornado Cash. *arXiv preprint arXiv:2201.06811*, 2022. URL: https://arxiv.org/abs/2201.06811.

[48] Uğur Şen, Sergei Tikhomirov, Sylvain Delhomme, Nadeem Bhati, and Cyprien Grau. RLN transaction submission microbenchmark artifact, 2026. Accessed: 2026-08-04. URL: https://github.com/sydhds/gasless_rln_impl.