

The Illusion of Payload Blindness: Why MEV Mitigations Fail in Practice

Simona Češková
Brno University of Technology
Brno, Czech Republic

Martin Perešíni
Brno University of Technology
Brno, Czech Republic

Ivan Homoliak
Brno University of Technology
Brno, Czech Republic
Slovak University of Technology in
Bratislava
Bratislava, Slovakia

Abstract

In public blockchains, the ability to order visible transactions allows block producers to extract value from users, a problem known as Maximal Extractable Value (MEV). Proposer-Builder Separation has turned that right into a formalized auction, and the leading protocol-level mitigations answer it with payload blindness: encrypted mempools, verifiable permutations, and block delay hide or scramble transactions before the order is fixed. Every such mitigation carries a security argument that assumes an adversary who learns nothing before the order commits. Deployed systems face a different adversary, and this paper measures the cost of that difference. We evaluate the three families of mitigations against a single adversary hierarchy, using closed-form analysis and simulations calibrated to Ethereum. Against a fully blind adversary, the outcome follows from the definitions: encryption with a permutation removes the attack classes that depend on transaction content, and block delay increases extraction rather than reducing it. Once that idealization is dropped, deployment problems dominate. The unencrypted envelope identifies the sender, so past behavior restores a usable guess at hidden content. Optional encryption gives builders a reason to starve the transactions it protects, and voluntary adoption then collapses. Key rotation turns a cryptographic parameter into a source of queueing congestion. Suppressing extraction, finally, removes far more producer revenue than it returns to users in lower fees, so the cost of every mitigation falls on the party that controls its deployment. The binding obstacles to MEV-free blockchains are therefore liveness and deployment incentives rather than cryptography.

Keywords

Maximal Extractable Value, Proof-of-Stake, Encrypted mempool, Transaction ordering, Metadata leakage, Proposer-Builder Separation, Deployment incentives, Ethereum

1 Introduction

In public blockchains, the right to order visible transactions allows block producers to extract significant value from users. This phenomenon, known as Maximal Extractable Value (MEV), pays searchers and builders at the expense of users, who suffer worse execution prices, higher priority fees, and reverted transactions [12, 61]. While MEV exists in any network where payloads are public before ordering, Ethereum’s move to Proof-of-Stake (PoS) fundamentally

changed the economics of this extraction. Proposer-Builder Separation (PBS) [32], deployed out of protocol as MEV-Boost [27], turned the right to order transactions into a formalized revenue stream, and builder auctions now assemble close to 90% of Ethereum blocks [46, 57].

Three families of protocol-level mitigations address the problem. Encrypted mempools hide transaction payloads until the order is fixed, either through a threshold committee or through a self-decrypting time-lock construction [8, 22, 37, 47, 48, 51, 65]. A verifiable permutation reorders the candidate set so that no party controls the position [38, 40]. Block delay separates commitment from execution across slots [4, 34], and rollups move ordering off Layer 1 [25]. Each family arrives with a security argument. The families are rarely compared under a single adversary, and their effects on liveness, privacy, fees, and deployment receive less attention than their ordering guarantees.

The addressed gap. An ordering guarantee is proved against an idealized adversary that learns nothing before the order commits. A deployed mechanism faces a different adversary that reads the unencrypted envelope required by gossip and fee validation, decides whether to include an encrypted transaction at all, and decides whether to run the mitigation in the first place. We argue that the interesting failures of MEV mitigations occur at these three interfaces rather than within the cryptography. We therefore index our results by the adversary they assume, and we label every claim either as a consequence of the model or as a measured outcome. The first kind is a reference point rather than a discovery: payload blindness removes content-dependent attacks because the attacker cannot read the content, and uniform permutation removes position-dependent attacks because no party chooses position. The second kind measures what happens once the attacker’s idealization is dropped, and two of those results overturn remedies or optimizations that the designs we evaluate recommend. One boundary is worth stating early: public-state arbitrage exploits existing on-chain data rather than unconfirmed pending transactions, so no amount of payload hiding can prevent it [33, 49]. Since recent measurements place this cross-venue arbitrage at the center of extracted value [61], a perfectly deployed encrypted mempool would address only a minority of the total extraction, though it successfully removes the attack classes that directly target retail users.

Scope. Every quantitative claim holds inside the abstractions of Section 3: constant-product AMM execution, Poisson arrivals with deterministic batch service, an unbiased randomness beacon, and the calibration sources of Table 1. We calibrate to Ethereum because

Declaration on Generative AI. The authors used generative AI (ChatGPT-5) tools as writing aids. All ideas, experiments, and results are the authors’ own, and the authors take full responsibility for the content.

it is the largest deployment, and the findings apply to any leader-based PoS chain with builder-style block construction.

Contributions.

- (i) We quantify what the unencrypted sender envelope costs the blindness argument on which encrypted-mempool designs rest, we show that the address rotation they propose as the remedy repairs it only for atypical senders, and we identify the observable level of per-sender behavioral concentration below which the leak disappears.
- (ii) We show that optional encryption is a liveness trap in which builders starve encrypted transactions and voluntary adoption collapses, and we isolate the design choice that causes this, explaining why opt-in private relays do not suffer the same fate.
- (iii) We locate the binding obstacle to deployment: a reward-only subsidy must price an unbounded per-slot quantity that a rule enforced by fork choice never has to price.
- (iv) We compare the three families against a single adversary hierarchy, stating which claims are definitional, and show that builder concentration in replicator models of PBS is a trajectory rather than an equilibrium, so a concentration measurement alone cannot identify the state of the market.

Prior work. Earlier studies examine a single mechanism or address a different question, and we are not aware of prior work that evaluates all three families against a shared adversary and reports their liveness and deployment costs together. BlindPerm permutes historical traces without modeling threshold cryptography or builder strategy [37], empirical replays test only whether permuted blocks remain valid [18], commit-and-reveal models omit mempool dynamics [4], enshrined-PBS analyses keep the mempool in the clear [60], and agent-based work targets execution-ticket markets [53] or validator geography [63].

2 Background and Related Work

2.1 Canonical MEV Classes

We follow existing MEV taxonomies [34, 41, 64] and use four classes. A victim transaction T_v trades against an automated market maker (AMM), and an attacker submits T_a , or the pair T_{a_f} and T_{a_b} . *Front-running* places T_a immediately before T_v and moves the execution price against the victim; when the attacker instead forces T_v to revert, the same primitive becomes suppression [17]. *Back-running* places T_a immediately after T_v and rebalances the pool without changing the victim’s price. *Sandwiching* encloses T_v between T_{a_f} and T_{a_b} and extracts the slippage tolerance the victim granted [42]. *Arbitrage* aligns prices across venues, and it differs from the other three in one respect that governs the rest of this paper: its trigger is public chain state rather than the content of a pending transaction, so hiding the mempool does not remove it [33, 49].

2.2 Mitigation Families

PBS and user-level defenses. PBS splits block construction between builders and proposers [32], and MEV-Boost instantiates it out of protocol. PBS redistributes extraction to proposers instead of removing it, and it concentrates the builder market [7, 46, 57, 62], which motivates enshrined variants [60]. Users also have defenses

today that need no protocol change: private relays [26], order-flow auctions [16], profit-preserving protected bundles [5], and client-side slippage limits. Two properties separate these from the mechanisms we study. They keep a protected transaction out of the public mempool instead of constraining how a block is ordered, so they move the ordering right to a different holder rather than removing it, and they replace a protocol guarantee with trust in a relay or a solver.

Encrypted mempools. These hide payloads until the order commits, and they differ in whom the user must trust. Threshold designs distribute decryption across a validator committee or a keyper set, so confidentiality is preserved even if fewer than a threshold of members collude [8, 20, 37, 47, 51, 65]. Batched threshold encryption reduces the per-batch setup cost [14, 15], recent schemes cut that cost to quasilinear and add stake-weighted committees [1] or remove the per-batch setup entirely [10], and traceable variants identify a misbehaving member [13]. LUCID (EIP-8184) [22] packages sealed transactions in a scheme-agnostic way and composes with Fork-Choice Enforced Inclusion Lists (FOCIL, EIP-7805) [55]; both are drafts under active revision, and LUCID has drawn a substantive critique of its reveal optionality [35], so we treat neither as settled. A second construction removes the committee because delay encryption and verifiable delay functions make a ciphertext self-decrypt after a fixed amount of sequential work [11, 48]; its trust assumption is a hardware assumption, and an adversary with faster sequential hardware than the assumed calibration can decrypt early and leave no on-chain trace. Encryption removes content-dependent attacks by construction, and it introduces key-rotation overhead, committee liveness risk, and metadata leakage. Independent deanonymization results confirm that the metadata channel is practical rather than hypothetical [9, 59].

Verifiable permutation. Permutation reorders the candidate set using verifiable randomness or a fair-ordering protocol [2, 37, 38, 40, 47], thereby removing deterministic sandwiching. It cannot stop a builder from omitting a transaction, and it cannot deny a producer risk-free arbitrage on an arbitrary candidate set [24]. Strict order fairness also admits Condorcet cycles [56], so we analyze uniform random permutation instead.

Block delay. Block delay separates the slot in which a transaction commits from the slot in which it executes, so a sender publishes it in block N and it becomes executable only in block $N + \Delta$ for some $\Delta \geq 1$ [4, 34], which is meant to remove the real-time latency race. We evaluate delay as a standalone ordering rule, in which the transaction body is public at commit time and only execution is deferred; delay combined with a hidden payload is a different mechanism and has its own row in Table 2. It has not been deployed on Ethereum, and we include it because it remains the reference commit-and-reveal baseline in the mitigation literature.

3 Methodology

3.1 Models and Their Scope

We use one adversary hierarchy and four models of varying fidelity. Table 1 lists the model, horizon, and replicate count for each reported number. We derive ordering probabilities in closed form and

Table 1: Provenance of every reported experiment. DES is the transaction-level discrete-event simulator, and the remaining rows use purpose-built side models.

Experiment	Model	Horizon	Reps	Calibration
Ordering probs. (§4.1)	analytic	2×10^4 draws	n/a	analytic
Builder arbitrage (§4.2)	DES	300 blocks	30 seeds	RPC, Dune
Block delay (§4.3)	DES	100 slots	25 seeds	RPC trace
Optional enc. + premium (§5.2)	DES	1k to 1.5k slots	20 seeds	RPC fees
Profiling + sweep (§5.1)	resampling	150 swaps	20 seeds	personas
Rotation + load sweep (§5.3)	queue	256 slots	30 seeds	assumed ρ
Fee market (§6.1)	auction	5,000 slots	30 seeds	RPC, EigenPhi
Builder bimodality (§6.3)	replicator	6.4×10^4 slots	15 seeds	relay shares
Builder horizon (§6.3)	replicator	$\leq 3.2 \times 10^4$ slots	25 seeds	relay shares

validate them against Monte Carlo simulation with 20,000 draws per point and 95% percentile bootstrap intervals from 1,000 resamples. A slot-level discrete-event simulator (DES) models the 12-second Ethereum slot and represents user submission, mempool propagation, builder optimization, searcher competition, and validator execution. It supports four ordering modes: priority-fee baseline, verifiable uniform permutation, block delay, and mandatory threshold encryption with permutation. In the encrypted mode, the block closes, and the order is fixed over the ciphertexts before decryption, so no party can insert a transaction after a payload becomes readable. This sequencing is part of the mechanism definition assumed by Table 2. The execution layer models constant-product pools ($x \cdot y = k$, as in Uniswap V2) and omits general EVM callbacks, so our quantitative results reflect AMM swap traffic rather than arbitrary contract logic. The DES produces the builder-arbitrage, block-delay, and optional-encryption results. Four further results use purpose-built side models: a slot-quantized queue for key rotation, a single-round pay-as-bid auction for the fee market, a replicator model for builder concentration, and a resampling experiment over a synthetic trader population for sender profiling.

Mainnet calibration. We fit arrival rates and priority-fee distributions to a public Flashbots RPC trace of 100 consecutive blocks (heights 24,898,874 to 24,898,973, April 2026). We derive venue and pair distributions from Dune Analytics aggregates of long-run DEX swap volume, and we derive builder shares from public relay data [19, 21, 28, 57]. The trace fixes input parameters only; it is a snapshot, not an evidence base, and we do not extrapolate any result beyond those 100 blocks. We sweep rather than assume parameters that it does not determine, including the correcting-flow fraction q , the delay Δ , the deferral fraction f , the key window W , the queue load ρ , and the order-flow feedback α . Base fees differ between the calibration window and historical congested periods, because blob transactions reduced them after EIP-4844. This affects absolute revert costs in fiat terms, not our reported quantities, which depend on priority-fee differences and relative extraction margins.

3.2 Adversary Capability Hierarchy

Mechanism security and deployed security differ because deployed systems leak information that idealized models assume away, so we index every result by the adversary it assumes. At *Level 0*, the adversary is content-blind: payloads, methods, and metadata stay hidden until the order is fixed. This defines the theoretical limit in Section 4, and no deployed design reaches it, because outer envelopes must expose routing and fee fields. At *Level 1*, the adversary

is metadata-aware and reads the unencrypted envelope, which carries the sender address, gas limit, and fee fields, and infers the venue, pair, and direction from behavioral history (Section 5.1). At *Level 2*, an integrated builder with exclusive assembly rights inserts its own transactions or omits others, with no competing searchers (Section 4.2). At *Level 3*, parties defect strategically: users choose between the encrypted and the plaintext path (Section 5.2) or defer submission near key-rotation boundaries (Section 5.3), and producers choose whether to run the mitigation at all (Section 6).

Exposure of price-correcting flow. Builder arbitrage depends on how much price-correcting order flow reaches the block, and two causes reduce that flow: a coalition of builders can filter arbitrage transactions, and ordinary users can trade in directions that widen a pool imbalance. Our simulator exposes one parameter for both, written $q \in [0, 1]$, where a fraction $1 - q$ of user flow moves prices toward cross-venue parity, and we sweep q over $[0.1, 0.7]$, whose upper end is the pessimistic setting that a censoring coalition creates. We do not claim that q measures observed mainnet censorship, and we use it only to bound how builder arbitrage responds to the supply of correcting flow.

4 The Idealized Reference Point

This section fixes the reference point against which Section 5 measures deployment costs, and we state at the outset which of its claims are definitional. Under Level-0 blindness, an attacker cannot read a pending payload, so no attack that requires knowing the payload can succeed, and under uniform permutation, no party chooses a position, so no attack that requires a chosen position can succeed. Those two statements are consequences of the model. Two results here are not: the arbitrage a builder retains under permutation alone, and the direction in which block delay moves extractable value.

4.1 Ordering Probabilities under Permutation

Take a block of B candidates under a uniformly random permutation, with N searchers competing for one opportunity, so all $B!$ orderings are equally likely. Public-state arbitrage goes to the searcher transaction that lands earliest, giving $P_A(N) = 1/N$; an adjacency-dependent attack needs the attacker beside the victim inside the $(N + 1)$ -element ordered subset of victim and searchers, giving $P_{FB}(N) = 1/(N + 1)$; and a single-attacker sandwich needs one of the $3! = 6$ orderings of T_{af}, T_v, T_{ab} , so $P_{SW} = 1/6$, which account nonces raise to $1/3$. Monte Carlo simulation reproduces all four, with the largest gap of 0.0059 at the unrestricted sandwich with $N = 1$, which is 2.2 standard errors, so we report it rather than call the agreement exact. Both sandwich variants fall below 1% at $N \geq 10$. Permutation, however, does not deter spam, because a searcher that submits k transactions rather than one raises its chance of landing first to $k/(N + k - 1)$ at a base fee F each. Its expected return $kV/(N + k - 1) - kF$ for opportunity value V is maximized at $k^* = \sqrt{(N - 1)V/F} - (N - 1)$, which at $N = 10$ and typical DEX values ranges from about 1 to about 21, so splitting pays wherever competition is thin or the opportunity is large.

4.2 Builder Arbitrage: What Remains

Permutation shuffles the contents of a block whose candidate list the builder still controls, allowing the builder to insert its own arbitrage transaction sized to the cross-pool spread left open by the shuffled user flow. Across 30 seeds of 300-block runs, that transaction clears its own gas cost in 85.2% of permuted blocks at $q = 0.5$ (95% CI [84.3, 86.1], between seeds), rising to 94.9% at $q = 0.7$ and falling to 52.3% at $q = 0.1$, the monotone response to the supply of price-correcting flow defined in Section 3.2 (Figure 1a).

Adding mandatory encryption changes the picture (Figure 1b). Payload blindness alone, with no competing searchers, reduces clearance to 62.2%, because the builder must size its trade without seeing the flow it trades against, and ten blind searchers reduce clearance to 9.1% (95% CI [8.5, 9.7]). The mechanism is competition and not infeasibility: blind searchers bid on the same public spread, they cannot see each other’s sizes, and the losers pay gas on reverted transactions, which gives a mean profit per block of $-\$301$ at ten searchers under unshaded bidding and makes part of the loss social rather than a transfer. Rational searchers respond by shading bids or entering at random, and the cross-pool price gap still closes, so we mark this cell as *dissipated* rather than *prevented* in Table 2.

4.3 Encryption Alone, and Why Delay Moves the Wrong Way

Encryption removes the content-dependent classes at Level 0 and leaves public-state arbitrage untouched. A blind attempt at front-running or sandwiching requires four independent successes: the attacker must guess the content, land in the right position, execute validly, and clear costs. Only the first is specific to encryption, and guessing the content means guessing the venue, the pair, and the direction. Counting each AMM version as a separate venue, the leading venue accounts for about half of all swaps, and the second for about a third, whereas pair concentration within a venue is much weaker because activity spreads across thousands of pools [19]. Counting all versions of a single protocol as a single venue raises concentration sharply, so the figure depends on a modeling choice that we state explicitly rather than leave implicit. We fix the content-blind baseline on the same population we later attack, so the two are comparable: on that population, an attacker restricted to population-level frequencies achieves a joint accuracy of $P_C = 0.047$. The baseline covers blind attempts to act *before* a pending transaction, and it does not cover back-running, which needs no payload guess because a back-runner reacts to the state the victim has already produced. Back-running is therefore the only rational unblinded strategy against payload encryption used on its own.

Block delay is the one mechanism whose sign we can determine rather than assume. Deferring execution by Δ slots widens the window in which external prices move away from pool reserves, and under a jump-diffusion price process, the resulting cross-pool arbitrage grows sharply. Mean extractable value per slot stays below one dollar at $\Delta = 1$ ($\$0.86$, 95% CI [0.39, 1.33]) and reaches $\$161$ at $\Delta = 3$ and $\$612$ at $\Delta = 5$ across 25 seeds, while reverts rise from near zero to 8.3% because user slippage bounds refer to a state that no longer holds at execution time. Growth is concentrated in the tail, and median extractable value stays at zero through $\Delta = 4$. Delay,

Table 2: Mitigation effectiveness by attack class, under the Level-0 adversary only. ✓ prevented within the model abstractions, ~ conditional, ✗ feasible. Suppression is front-running that forces the victim to revert. Every prevention mark assumes payload *and* metadata blindness.

Mechanism	Front-run	Suppress.	Back-run	Sandwich	Arbitrage
No mitigation	✗	✗	✗	✗	✗
Block delay	✗	✗	✗	✗	✗ ^a
Uniform permutation	~	~	✗	✓	✗
Payload encryption	✓	✓	✗	✓	✗
Encryption + permutation	✓	✓	✓ ^b	✓	~ ^c
Delay + encryption + permutation	✓	✓	✓ ^b	✓	~ ^c

^a Amplified rather than left unchanged. ^b In-block only, and only because the order is fixed over ciphertexts before decryption. ^c Dissipated, not prevented: blind competition removes searcher profit while the price gap still closes.

therefore, does not redistribute MEV; it manufactures it, and it does so in the one class that encryption cannot cover.

4.4 Summary

Table 2 collects the outcomes. Permutation alone removes sandwiching, weakens front-running, and leaves back-running intact because a back-runner profits from any position after the victim and can revert cheaply when it fails. Encryption combined with permutation is the minimal configuration that removes all three content-dependent classes, and it still does not remove arbitrage. Neither primitive stops back-running alone, and the combination stops it only under the sequencing stated in Section 3.1, where the order is fixed across ciphertexts before any payload is readable, so no party can react to the victim within the same block. Reaction in a later slot is public-state arbitrage rather than back-running, and it appears in the last column. Two classes we do not simulate follow the same division: just-in-time liquidity provision needs an identified victim and adjacency, and oracle front-running survives every mechanism here.

5 Failures Outside the Idealization

The reference point of Section 4 rests on three assumptions: the adversary learns nothing before the order commits, every user takes the protected path, and some party is willing to run the mechanism. This section drops the first two, and Section 6 drops the third. None of the results below follow from the definitions, and two overturn remedies or optimizations that the designs we evaluate recommend.

5.1 Defeating the Blindness Argument

Encrypted-mempool designs leave the outer envelope in the clear because gossip and fee validation require the sender address, gas limit, and fee fields, so the sender identity is public even when the payload is not, and a sender who trades consistently is predictable. The designs treat this residue differently, and none of them measure it: MEVade argues that blinding makes advance knowledge of a transaction impossible [47], F3B classifies sender-address leakage as a side channel enabling *speculative* front-running and proposes that a sender pay from a fresh address [65], and BlindPerm cites metadata leakage as a reason to add permutation on top of encryption [37]. Prior work has established that profiling Ethereum

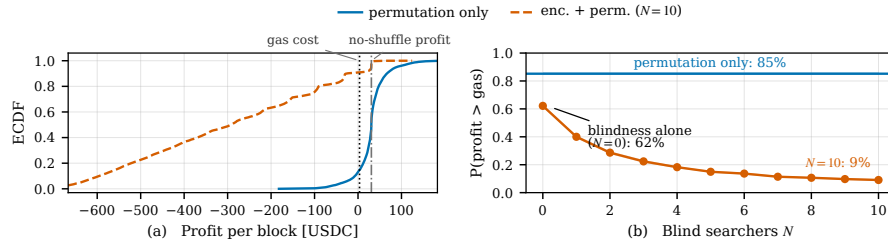


Figure 1: Builder arbitrage under permutation. (a) Profit per block, with gas cost and unshuffled profit marked. (b) Probability that profit exceeds gas cost against the number of blind searchers N , with between-seed intervals. The case $N = 0$ isolates blindness from competition.

senders is possible [9, 59]. What is new here is how much we recover from an encrypted mempool, and that the address remedy F3B proposes fails for exactly the senders it is meant to protect.

We measure the advantage on a synthetic population of seven trader personas, each with 20 senders, trading across five venues, seven pairs, and two directions, with 50 history swaps per sender. The adversary sees only pre-decryption features, the sender’s own historical distribution, and the public submission time; its predictor is the simplest available: it outputs the sender’s most frequent historical label. Across 20 seeds, this trivial predictor attains 8.2 to 23.7 times the accuracy of uniform guessing (Table 3).

Two properties matter more than the magnitude. First, the advantage stems from persistent user heterogeneity rather than from a learned temporal pattern. Random forests and logistic regression over the same features score strictly below the mode predictor for every persona, with non-overlapping intervals, because they fit noise that the mode ignores. A designer, therefore, cannot dismiss this channel as an artifact of adversary sophistication. Second, every persona exceeds the content-blind matched baseline in Section 4.3, which is $P_C = 0.047$ on the same population, by factors ranging from 2.5 for unskilled retail senders to 7.2 for automated bots. The prevention marks of Table 2 hold under payload *and* metadata blindness, and they do not transfer to a deployment that publishes the sender.

How much does the result depend on the population? The personas are synthetic, and we drafted their weight vectors rather than fitting them, so the honest question is not whether these numbers describe Ethereum but at what level of behavioral concentration the leak appears at all. We sweep a tilt exponent over every sampled per-sender preference vector, from one that removes all per-sender preference to one that sharpens it, and index each resulting population by an observable statistic C , the mean realized share of a sender’s swaps that fall on that sender’s own most-used venue, which any operator can measure from public swap history today. The advantage over the content-blind baseline is 0.96 at $C = 0.24$, the floor produced by senders who choose venues uniformly at random, and rises monotonically to 1.04 at $C = 0.26$, 2.16 at $C = 0.41$, 3.57 at $C = 0.55$, and 5.78 at $C = 0.72$. This converts an assumption about a synthetic population into a falsifiable threshold on a measurable quantity and identifies the only safe population: one in which senders have no stable habits.

Does address rotation repair it? Rotation is the obvious user-side countermeasure, and it is cheap. We rerun the adversary while

Table 3: Adversary accuracy from pre-decryption metadata alone, under the mode predictor, with 95% intervals within ± 0.02 . Joint accuracy is the product of the three marginals. The population prior is the content-blind baseline. The last two columns rotate the sender address every R swaps.

Persona	No rotation				Rotation	
	Venue	Pair	Dir.	Joint	$R=10$	$R=1$
Uniform guess	0.200	0.143	0.500	0.0143	n/a	n/a
Population prior	0.351	0.249	0.543	0.047	n/a	n/a
Automated bot	0.809	0.558	0.750	0.339	0.278	0.214
High-capital	0.629	0.422	0.752	0.200	0.149	0.119
Arbitrage optimizer	0.530	0.414	0.744	0.163	0.086	0.002
Trend follower	0.475	0.378	0.764	0.138	0.080	0.011
Casual trader	0.465	0.354	0.732	0.120	0.075	0.035
Institutional	0.449	0.360	0.738	0.119	0.079	0.054
Unskilled retail	0.445	0.353	0.743	0.117	0.065	0.013

each sender moves to a fresh address after every R swaps, so the rotation interval R is the number of swaps a sender makes from one address, and the adversary sees only the history of the current address and falls back to the population prior to when an address is new. Rotation helps, but unevenly (Table 3). Senders whose behavior differs from the population lose almost all of their exposure, and the arbitrage optimizer falls from 0.163 to 0.002 at a rotation interval of one swap per address ($R = 1$), far below the baseline, because the prior is actively wrong about it. Senders whose behavior matches the population keep most of theirs, and the automated bot falls only from 0.339 to 0.214, still 4.5 times the baseline. Rotation conceals which sender is acting and cannot conceal that a sender acts like the crowd, so protecting typical senders requires hiding the sender at the protocol level rather than the account level, for which stealth-address schemes such as ERC-5564 [58] are the concrete primitive.

5.2 Optional Encryption Is a Liveness Trap

BlindPerm [37] and MEVade [47] make encryption optional and permute only the blocks that contain an encrypted transaction, the first as a bandwidth optimization and the second as a fixed gas surcharge for blinding. Neither accounts for the negative externality that the choice creates on the builder. Accepting one encrypted transaction forces the whole block to be permuted, forfeiting the extraction margin on every plaintext transaction in it, so a profit-maximizing builder prefers blocks with no encrypted transactions whenever they carry extractable value.

The consequence is unequal inclusion at equal fees (Figure 2a). We report the *starvation differential*, the difference in the fraction of transactions evicted after a mempool lifetime of $H = 10$ slots,

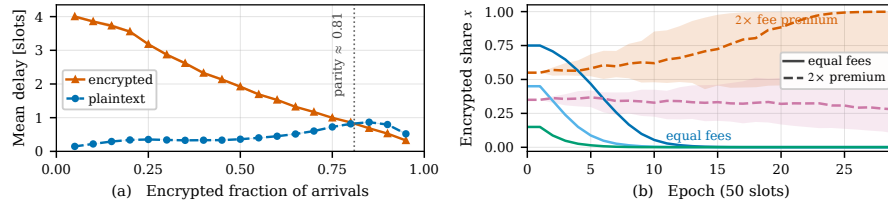


Figure 2: Optional encryption at residual leakage $\lambda = 0.1$. (a) Mean inclusion delay at equal fees, which reaches parity only near an encrypted share of 0.81. (b) Median encrypted share over 30 epochs. Solid curves use equal fees and collapse, and dashed curves add a premium of twice the plaintext tip.

over residual leakage $\lambda \in \{0, 0.1, 0.5, 0.9\}$ and encrypted share $p_{\text{enc}} \in \{0.05, 0.2, 0.5\}$ at 20 seeds per cell. At $\lambda = 0.1$, it runs from +51 to +63 percentage points across adoption shares, every 95% interval is within ± 0.02 , and it does not fall below +31 points even at $\lambda = 0.9$, where permutation is nearly worthless to the user. Users respond to that gap: when each best responds to the inclusion latency it observed, the encrypted share collapses to the floor from every initial share up to 0.85 (Figure 2b). Only an initial share of 0.95 survives, and it decays to a median of 0.32 rather than stabilizing, so optional encryption cannot bootstrap from any realistic starting point.

Is a fee premium the right answer? Charging more for encrypted transactions is defensible because encryption imposes additional costs on the network, and a protected user receives something of value. The premium works only in part. Over 20 premium levels at 20 seeds each, a premium of 1.8 to 2.4 times the plaintext tip equalizes eviction rates across the adoption shares we test, and it does not equalize latency. At adoption shares of 0.2 or below, the mean inclusion delay for encrypted transactions remains above that for plaintext even at 4.6 times the tip, because a builder that skips one keeps the option to extract from the rest of the block, and no tip removes that option. The premium also converts a trap into a coordination game with an unstable threshold near 0.5, so the outcome depends on where adoption starts, and neither outcome is a protocol guarantee.

What exactly causes the trap. The result is a property of one design choice and not of voluntary protection in general, because the externality exists only where a single ciphertext changes how the *rest* of the block is ordered. Designs that keep protected flow out of the public mempool without constraining anything else do not create it, which is why opt-in private relays and order-flow auctions have scaled to large volume without the collapse we measure [5, 16, 26]. The lesson is correspondingly precise: encryption must be mandatory whenever the ordering rule it triggers applies to the whole block, and an optional scheme survives only if it confines that constraint to the protected subset, which leaves plaintext flow unprotected. A live deployment of opt-in encrypted transactions exists on Gnosis Chain [30], and comparing realized inclusion latency there is the natural empirical test of this prediction.

5.3 Key Rotation Creates Congestion

Threshold committees change the key that their ciphertexts address on a schedule, and the schedules differ: MEVade derives a fresh committee key every slot [47], Ferveo runs its key generation once per epoch [8], and Shutter publishes a long-lived eon key and

releases only per-slot decryption keys against it, so its boundary arrives when the keyper set changes [20, 51]. Epochless constructions remove the boundary altogether [10, 15], so this result scopes to designs whose ciphertexts address a committee that changes, and we sweep the window W rather than adopt one schedule. A ciphertext addressed to an outgoing committee can be handled in three ways: `drop_resubmit` discards it and relies on the sender to retransmit, `overlap` keeps the outgoing keys valid for one further window, and `long_lived` keeps them valid indefinitely.

We model the mempool as a slot-quantized queue with Poisson arrivals and deterministic service at a rate of 200 transactions per slot. With no user reaction, all three strategies behave alike at offered load $\rho = 0.96$ (mean queue depth 5.7, peak 54.5). The difference appears once users behave strategically, because a user who knows a rotation is imminent gains nothing by submitting to it. At a deferral fraction of $f = 0.25$, deferred traffic returns as a synchronized burst of about 237 transactions at the start of the next window, mean queue depth rises to 66, peak depth to 219, and 99th-percentile latency to three slots (Figure 3a). Sweeping the load establishes that the boundary, rather than the load level, creates this congestion. At $\rho = 0.70$ the queue without deferral is empty in every slot of every seed, yet the same deferral raises peak depth to 101 and 99th-percentile latency to three slots, which that percentile reaches at every load from 0.70 to 0.99. The ratio of deferred to undeferred peak depth meanwhile falls from 94 at $\rho = 0.80$ to 2.1 at $\rho = 0.99$, so saturation hides the effect rather than causing it.

Window length trades bandwidth against exposure, and cumulative retransmissions over 256 slots fall from 1,415 at $W = 1$ to 50 at $W = 32$ (Figure 3b). The `overlap` strategy keeps the queue stable at any window length by making unincluded window- W ciphertexts readable in window $W + 1$, which exchanges boundary confidentiality for liveness. Commit-and-decrypt adds inclusion latency on top of this, turning a 12-second slot into a 24-second cycle whenever execution follows the commitment in the next slot. Same-slot packaging is the proposed answer, but it is not yet available: LUCID [22] separates payload commitment from key release; its baseline flow still executes in the following slot; and same-slot execution is an unadopted extension whose reveal-optionality has been challenged [35]. The latency cost is therefore a property of the packaging rather than of encryption, and it remains unsolved.

5.4 Timing Games

MEV accumulates with elapsed slot time, which rewards proposers that delay their proposals [29, 45, 50], and single-leader protocols

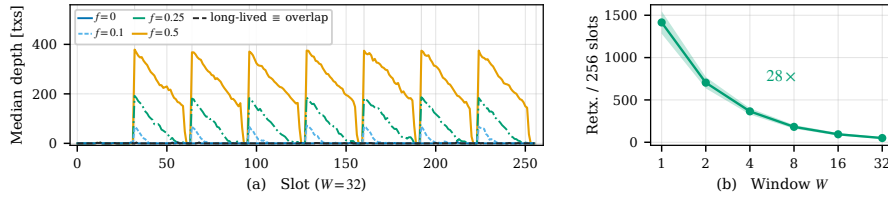


Figure 3: Key rotation at window boundaries. (a) Median queue depth per slot at key window $W = 32$ under `drop_resubmit`, for deferral fractions f . The dashed line is `long_lived` and `overlap`, which coincide. (b) Cumulative retransmissions against window length W with no deferral.

need large stake penalties to deter this, whereas multi-leader designs deter it through competition [3, 23, 39, 54]. Competition absorbs the incentive only while the late-to-early ratio remains below the number of leaders, and block delay raises that ratio to roughly 190 at three slots and 710 at five slots. Mandatory encryption with permutation flattens the curve instead of paying for it, so the design that removes MEV and the design that removes timing games are the same design.

6 Who Deploys a Mitigation?

The incentive gap. Mitigations that suppress MEV create a conflict between the users who benefit and the producers who deploy. Users gain from better execution prices and fewer reverted transactions, while producers lose the extraction margin, a direct source of their revenue. A design that relies on voluntary adoption, therefore, asks the party that loses from a mitigation to be the party that ships it, which is why the results below concern budgets and fork-choice rules rather than ciphertexts.

6.1 Who Pays and Who Saves

We evaluate a single-round pay-as-bid inclusion market at an offered load of $\rho = 0.96$. User tips follow a lognormal fitted to the median and 95th percentile of the calibration trace, MEV-bearing transactions arrive at the EigenPhi rate of 2.78 per slot and bid at the fitted 99th percentile so that they outbid ordinary users by construction, and the counterfactual deletes that flow while leaving the user tip distribution unchanged. Removing it reduces proposer priority revenue by 10.8%, reduces user tip costs by 0.5%, and raises the user inclusion rate by 0.5 percentage points. The magnitudes follow in part from those modeling choices and the sign does not: users gain mainly through better execution prices and fewer reverts, a channel producers do not monetize, whereas producers lose fee revenue they do monetize. The accounting is also incomplete: liquidity providers lose value to arbitrage that every mechanism here dissipates rather than prevents [43, 44], so a full welfare statement has three parties and no design we evaluate repairs the third.

6.2 Why a Subsidy Cannot Buy Compliance

Every mechanism in Table 2 that suppresses MEV shifts surplus from block producers to users, and the levers that deploy it are builder software, relay policy, validator defaults, and protocol upgrades, all of which producers control. The empirical record matches that asymmetry: MEV-Boost, the one ordering intervention adopted

at scale, raises producer revenue [46]; encrypted-mempool deployments run on operator subsidies [20, 30, 51]; and under 2% of proposers order blocks by fee alone with no external builder relationship, the strictest available proxy for a producer that gives up revenue on principle [52]. We formalize the gap with the producer utility decomposition $U = T + M + R$ from the responsive-consensus literature [3, 39], where T is observable protocol fees, M is unobservable extraction margin, and R is protocol-controlled transfers. A producer that runs mandatory encryption with permutation gives up priority tips ΔT_s and all but a residual leakage fraction λ of its margin M_s in slot s , and compliance is publicly verifiable because ciphertexts and permutation proofs are on the chain, whereas the quantity forgone is not.

PROPOSITION 6.1 (DEPLOYMENT BUDGET). *Let $M_s \geq 0$ have unbounded upper support and let $D_s = (1 - \lambda)M_s + \Delta T_s$ be the per-slot temptation to defect. A reward-only instrument pays a compliance subsidy S and leaves a non-compliant block canonical, so a defecting producer keeps D_s , incentive compatibility in slot s requires $S \geq D_s$, and compliance in every slot requires $S \geq \text{ess sup}_s D_s = \infty$. Adding a bond forfeited with detection probability p replaces this with $S + pB \geq D_s$, which likewise requires $B \rightarrow \infty$, because certain detection of the action does not bound the amount forgone. A fork-choice instrument instead makes a non-compliant block non-canonical. The deviation payoff is then zero for every realization of M_s , so incentive compatibility holds at $S_{IC} = 0$, and the protocol must fund only the mean gap $\mathbb{E}[D_s]$ to keep validators participating.*

The distinction is between pricing a deviation and removing it: a reward-only rule must be sized against the largest realization it will ever face, and a fork-choice rule never prices that realization, because the deviation produces no block. Unboundedness alone drives the impossibility, and the heavy right tail of per-slot MEV settles the separate question of what a finite budget buys: $\Pr[D_s > S]$ decays polynomially rather than exponentially under a Pareto tail. Hence, a subsidy sized for the median slot fails in exactly the slots worth defecting on. Two standard objections do not overturn this. Issuance yield raises the reward of the remaining validators until participation recovers, and risk pooling lets validators insure one another against the variance of M_s , but both settle for individual rationality within the validator set rather than for incentive compatibility for a slot. Only per-slot deterrence requires enshrinement in the style of FOCIL [55], and the same tension appears in transaction fee mechanism design, where no non-trivial mechanism is simultaneously good for users and incentive compatible for producers with private block values [6]. This states what would be required,

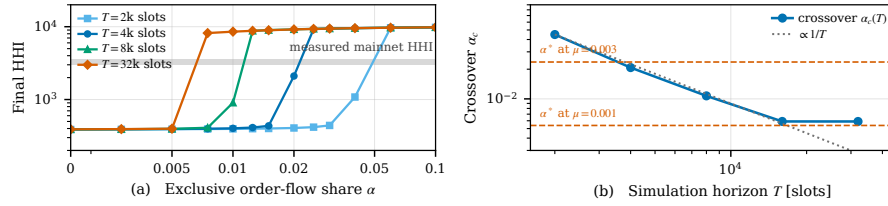


Figure 4: Builder concentration under exclusive order flow. (a) Final HHI against feedback strength α at four horizons, with the 2026 mainnet range shaded. Outcomes are bimodal, and the observed range lies between the modes. (b) Value of α at which the median HHI crosses that range, which falls as $1/T$ to a floor set by the entry rate μ .

not what is imminent: FOCIL is a draft deferred to the next scheduled upgrade, and no live proposal rejects plaintext blocks at the fork-choice level.

6.3 Who Holds the Levers

Encrypted mempools and permutation both operate on top of a builder market, so the state of that market conditions who can deploy anything, and public relay data show high concentration with unstable composition: across three snapshots of 2026 the Herfindahl-Hirschman Index (HHI) of builder block shares moved between roughly 2,950 and 3,550, while the ranks below the leader changed completely [28, 57, 62].

We ask whether a standard dynamic model reproduces such a state and extend replicator dynamics [36] to $n = 63$ builders across three hardware tiers. Each slot carries O MEV opportunities, a share $1 - \alpha$ of that flow is public, and the remaining share α is exclusive order flow routed in proportion to current market share [31, 46], so builder i observes $O_i = O[(1 - \alpha) + \alpha n x_i]$ for share x_i . Shares evolve under a logit update $x_i \leftarrow x_i e^{s F_i} / \sum_j x_j e^{s F_j}$ with selection strength $s = 0.05$ and an entry term of rate μ returning mass to the uniform distribution, and fitness is $F_i = \rho \pi_i - C_i$ for extracted value π_i , tier cost C_i , and retained share $\rho = 0.3$. With no exclusive flow, the market settles into a cost-stratified distribution at $\text{HHI} \approx 389$, and with exclusive flow, it has exactly two long-run states. Across 600 trajectories to a horizon of 64,000 slots, spanning a 15-fold range of α and a 33-fold range of μ , no run ended with an HHI between 1,000 and 7,000: every run ended either distributed, below 400, or near-monopoly, above 7,300 (Figure 4a).

Linearizing the logit map at the uniform-share vector explains why. A perturbation there grows by a factor $(1 - \mu)(1 + sh\bar{O}(\rho\bar{v} - c)\alpha)$ per slot, for tier hardware h , mean opportunity flow $\bar{O}$, mean value per opportunity $\bar{v}$, and marginal extraction cost c , so the uniform state is unstable for any α above a critical value that scales linearly with the entry rate and inversely with selection, retention, and opportunity flow, which means that entry sets the threshold and feedback does not. The simulated threshold sits an order of magnitude below that bound, at $\alpha^* \approx 0.005$ for $\mu = 10^{-3}$ against a predicted 0.061, because the market starts stratified by tier and per-slot extraction noise seeds instability early. Raising μ to 10^{-2} leaves no tested α that consolidates the market, which is the predicted linear scaling. Intermediate concentration appears only in transit, so the apparent critical strength depends on the horizon (Figure 4b), falling from $\alpha = 0.046$ at 2,000 slots to 0.0061 at 16,000 slots before it stops. An HHI near 3,000 is thus not a stable interior state of this model class. It is consistent both with a market moving toward consolidation and with a stabilizing force the model omits, such

as proposer multi-homing or capacity limits at the top builders. Deciding between the two requires measuring α and μ directly, which no public dataset reports.

7 Limitations

Four limits bound how far the results travel. *Coverage*. The simulator executes AMM swaps and not general EVM state, and it omits network topology, pre-image attacks, and beacon grinding, so our magnitudes describe DEX traffic, and the extended classes of Section 4.4 are argued rather than measured. *Calibration*. The 100-block trace fixes arrival and fee parameters only. We did not test whether another window shifts the fitted distributions, and we sweep the parameters that the trace does not determine. *Modeling choices*. The fee-market result depends on placing MEV bids at the fitted 99th percentile and on holding the user tip distribution fixed, so only its sign is robust, and the builder model is a learning dynamic rather than an equilibrium model of a first-price auction, which is why we use it only negatively. *Population*. The traders of Section 5.1 are synthetic, and we drafted their persona weights with the assistance of a generative model rather than fitting them to chain data, so we report the concentration sweep beside the point estimates: the threshold on C is the claim we ask readers to test, and the per-persona magnitudes illustrate one population at $C = 0.55$. Fitting personas to labeled searcher addresses and to per-wallet venue loyalty in public swap history is the obvious next step and requires no encrypted mempool. We package all simulators, sweeps, outputs, and figure generators for artifact review, with fixed seeds and one entry point per row of Table 1.

8 Conclusion

We evaluated encrypted mempools, verifiable permutation, and block delay against a single adversary hierarchy, and we labeled every claim as either a consequence of the model or a measured outcome. We built a slot-level simulator of Ethereum block production, three side models of key rotation, the fee market, and the builder market, and a resampling attack on pre-decryption meta-data. The definitional part of the answer is short: under a fully blind adversary, encryption with a permutation removes the three content-dependent MEV classes and dissipates the fourth, and block delay increases extraction rather than reducing it. The measured part is the contribution, and each result states a condition that a deployment must meet. *The sender must be hidden together with the payload*, because the envelope publishes the sender, and habitual behavior alone restores a usable guess at hidden content. *Encryption must be mandatory wherever its ordering rule applies to*

the whole block, because that coupling lets a builder keep the extraction margin of a block by declining one ciphertext, and voluntary adoption then collapses. *Deployment must be enshrined rather than subsidized*, because a reward-only instrument must be sized against an unbounded per-slot temptation that a fork-choice rule never prices. The takeaway is a change in agenda: block delay leaves the design space; payload blindness holds only where the attack needs the payload; and cross-venue arbitrage now dominates measured extraction [61]. An MEV-free chain is therefore not on offer, and what remains is blocked by liveness and deployment incentives rather than by cryptography.

References

- [1] Amit Agarwal, Kushal Babel, Sourav Das, Babak Poorebrahim Gilkalaye, Arup Mondal, Benny Pinkas, Peter Rindal, and Aayush Yadav. 2025. Weighted Batched Threshold Encryption with Applications to Mempool Privacy. IACR Cryptology ePrint Archive, Paper 2025/2115. Accepted, IEEE Symp. on Security and Privacy (SP) 2026, Distinguished Paper Award (page numbers pending IEEE Xplore indexing). <https://eprint.iacr.org/2025/2115>
- [2] Zeinab Alipanahloo, Kaiwen Zhang, and Emmanuel Awosika. 2024. Quantifying Fairness Granularity as a Fair Ordering Policy Towards MEV Mitigation for Rollups. In *2024 IEEE Int. Conf. on Blockchain (Blockchain)*. IEEE, Piscataway, NJ, USA, 209–218. doi:10.1109/Blockchain62396.2024.00035
- [3] Kaya Alpturk, Kushal Babel, and Aditya Saraf. 2025. Timing Games in Responsive Consensus Protocols. arXiv:2510.25144 [cs.GT].
- [4] Benjamin Appiah, Daniel Comney, Winful Bagyl-Bac, Laurene Adjei, and Ebenezer Owusu. 2025. Game-Theoretic Analysis of MEV Attacks and Mitigation Strategies in Decentralized Finance. *Analytics* 4, 3 (2025), 23. doi:10.3390/analytics4030023
- [5] Kushal Babel, Nerla Jean-Louis, Yan Ji, Ujval Misra, Mahimna Kelkar, Kosala Yapa Mudiyanse, Andrew Miller, and Ari Juels. 2026. PROF: Protected Order Flow in a Profit-Seeking World. In *2026 IEEE 11th European Symp. on Security and Privacy (EuroSecP)*. IEEE, Piscataway, NJ, USA, 398–418. Preprint: arXiv:2408.02303. doi:10.1109/EuroSP68448.2026.00034
- [6] Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. 2024. Transaction Fee Mechanism Design in a Post-MEV World. In *6th Conf. on Advances in Financial Technologies (AFT '24) (LIPIcs, Vol. 316)*. Schloss Dagstuhl, Dagstuhl, Germany, 29:1–29:24. doi:10.4230/LIPIcs.AFT.2024.29
- [7] Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. 2025. Centralization in Block-Building and Proposer-Builder Separation. In *Financial Cryptography and Data Security (FC 2024), Part I (LNCS, Vol. 14744)*, Jeremy Clark and Elaine Shi (Eds.), Springer, Cham, Switzerland, 331–349. doi:10.1007/978-3-031-78676-1_19
- [8] Joseph Bebel and Dev Ojha. 2022. Ferveo: Threshold Decryption for Mempool Privacy in BFT Networks. IACR Cryptology ePrint Archive, Paper 2022/898. <https://eprint.iacr.org/2022/898>
- [9] Ferenc Bérces, István András Seres, András A. Benczúr, and Mikera Quintyne-Collins. 2021. Blockchain is Watching You: Profiling and Deanonimizing Ethereum Users. In *2021 IEEE Int. Conf. on Decentralized Applications and Infrastructures (DAPPS)*. IEEE, Piscataway, NJ, USA, 69–78. doi:10.1109/DAPPS52256.2021.00013
- [10] Jan Bormet, Sebastian Faust, Hussien Othman, and Ziyang Qu. 2025. BEAT-MEV: Epochless Approach to Batched Threshold Encryption for MEV Prevention. In *34th USENIX Security Symposium (USENIX Security '25)*. USENIX Association, Berkeley, CA, USA, 3457–3476.
- [11] Jeffrey Burdges and Luca De Feo. 2021. Delay Encryption. In *Advances in Cryptology – EUROCRYPT 2021 (LNCS, Vol. 12696)*. Springer, Cham, Switzerland, 302–326. doi:10.1007/978-3-030-77870-5_11
- [12] Agostino Capponi, Ruizhe Jia, and Kanye Ye Wang. 2025. Maximal Extractable Value and Allocative Inefficiencies in Public Blockchains. *Journal of Financial Economics* 172 (2025), 104132. doi:10.1016/j.jfineco.2025.104132
- [13] Anirban Chakrabarti, Monosij Maitra, Arup Mondal, and Kushaz Sehgal. 2025. Silent Threshold Traitor Tracing and Enhancing Mempool Privacy. In *Proc. 2025 ACM SIGSAC Conf. on Computer and Communications Security (CCS '25)*. ACM, New York, NY, USA, 1769–1783. doi:10.1145/3719027.3765099
- [14] Arka Rai Choudhuri, Sanjam Garg, Julien Piet, and Guru-Vamsi Policharla. 2024. Mempool Privacy via Batched Threshold Encryption: Attacks and Defenses. In *33rd USENIX Security Symposium*. USENIX Association, Berkeley, CA, USA, 3513–3529. Preprint: IACR ePrint 2024/669.
- [15] Arka Rai Choudhuri, Sanjam Garg, Guru-Vamsi Policharla, and Mingyuan Wang. 2025. Practical Mempool Privacy via One-time Setup Batched Threshold Encryption. In *34th USENIX Security Symposium*. USENIX Association, Berkeley, CA, USA, 3477–3495.
- [16] CoW DAO. 2026. MEV Blocker. <https://mevblocker.io/>. Order-flow auction endpoint with searcher rebates. Accessed: 2026-08-08.
- [17] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In *2020 IEEE Symp. on Security and Privacy (SP)*. IEEE, Piscataway, NJ, USA, 910–927. doi:10.1109/SP40000.2020.00040
- [18] Jan Droll. 2025. Permuting Transactions in Ethereum Blocks: An Empirical Study. arXiv:2509.25415 [cs.DC].
- [19] Dune Analytics. 2026. Dune: Onchain Data Platform. <https://dune.com/>. Accessed: 2026-06-12.
- [20] Stefan Dziembowski, Sebastian Faust, and Jannik Luhn. 2024. Shutter Network: Private Transactions from Threshold Cryptography. IACR Cryptology ePrint Archive, Paper 2024/1981. <https://eprint.iacr.org/2024/1981>
- [21] EigenPhi. 2026. EigenPhi: MEV Analytics Platform. <https://eigenphi.com/>. Accessed: 2026-06-12; the analytics service was discontinued in June 2026.
- [22] Anders Elowsson, Justin Florentine, and Julian Ma. 2026. EIP-8184: LUCID Encrypted Mempool. Ethereum Improvement Proposals, no. 8184, status: Draft. <https://eips.ethereum.org/EIPS/eip-8184>. Created: 2026-03-04. Accessed: 2026-08-04.
- [23] Fatima Elsheimy, Ioannis Kaklamanis, Sarisht Wadhwa, Charalampos Papamantou, and Fan Zhang. 2026. Censorship Resistance vs Throughput in Multi-Proposer BFT Protocols. IACR Cryptology ePrint Archive, Paper 2026/126. <https://eprint.iacr.org/2026/126>
- [24] Matheus V. X. Ferreira and David C. Parkes. 2023. Credible Decentralized Exchange Design via Verifiable Sequencing Rules. In *Proc. 55th Annual ACM Symp. on Theory of Computing (STOC '23)*. ACM, New York, NY, USA, 723–736. doi:10.1145/3564246.3585233
- [25] Christof Ferreira Torres, Albin Mamuti, Ben Weintraub, Cristina Nita-Rotaru, and Shweta Shinde. 2024. Rolling in the Shadows: Analyzing the Extraction of MEV Across Layer-2 Rollups. In *Proc. 2024 ACM SIGSAC Conf. on Computer and Communications Security (CCS '24)*. ACM, New York, NY, USA, 2591–2605. doi:10.1145/3658644.3690259
- [26] Flashbots. 2026. Flashbots Protect. <https://docs.flashbots.net/flashbots-protect/overview>. Private transaction relay with optional MEV-Share refunds. Accessed: 2026-08-08.
- [27] Flashbots. 2026. MEV-Boost Overview and Flashbots Documentation. <https://docs.flashbots.net>. Accessed: 2026-06-12.
- [28] Flashbots. 2026. Relayscan: MEV-Boost Relay and Builder Statistics. <https://www.relayscan.io/overview?t=7d>. 7-day builder market share, displayed directly on the page (accessed 2026-08-04, $n = 45,886$ blocks): Titan 52.90%, Quasar 20.34%, BuilderNet 13.06%, Eureka 9.53%, others 4.17%; implied HHI ≈ 3477 (computed by the authors from the displayed percentages over MEV-Boost-delivered blocks; relayscan itself does not report an HHI figure).
- [29] Robin Fritsch, Maria Inês Silva, Akaki Mamagishvili, Benjamin Livshits, and Edward W. Felten. 2026. MEV Capture Through Time-Advantaged Arbitrage. In *Mathematical Research for Blockchain Economy (MARBLE 2025) (Lecture Notes in Operations Research)*. Springer, Cham, Switzerland, 1–22. doi:10.1007/978-3-032-13377-9_1
- [30] Gnosis. 2026. Shutterized Gnosis Chain. <https://docs.gnosischain.com/shutterized-gc/>. Opt-in encrypted transaction path on a live network. Accessed: 2026-08-08.
- [31] Tivas Gupta, Mallesh M. Pai, and Max Resnick. 2023. The Centralizing Effects of Private Order Flow on Proposer-Builder Separation. In *5th Conf. on Advances in Financial Technologies (AFT '23) (LIPIcs, Vol. 282)*. Schloss Dagstuhl, Dagstuhl, Germany, 20:1–20:15. doi:10.4230/LIPIcs.AFT.2023.20
- [32] Lioba Heimbach, Lucianna Kiffer, Christof Ferreira Torres, and Roger Wattenhofer. 2023. Ethereum’s Proposer-Builder Separation: Promises and Realities. In *Proc. 2023 ACM Internet Measurement Conf. (IMC '23)*. ACM, New York, NY, USA, 406–420. doi:10.1145/3618257.3624824
- [33] Lioba Heimbach, Vabuk Pahari, and Eric Schertenleib. 2024. Non-Atomic Arbitrage in Decentralized Finance. In *2024 IEEE Symp. on Security and Privacy (SP)*. IEEE, Piscataway, NJ, USA, 3866–3884. doi:10.1109/SP54263.2024.00256
- [34] Lioba Heimbach and Roger Wattenhofer. 2022. SoK: Preventing Transaction Reordering Manipulations in Decentralized Finance. In *Proc. 4th ACM Conf. on Advances in Financial Technologies (AFT '22)*. ACM, New York, NY, USA, 47–60. doi:10.1145/3558535.3559784
- [35] Gottfried Herold. 2026. A Criticism of LUCID and Encryption-Scheme-Agnostic Encrypted Mempool Designs. <https://ethresear.ch/t/a-criticism-of-lucid-and-encryption-scheme-agnostic-encrypted-mempool-designs/25210>. Ethereum Research forum, 2026-06-16. Accessed: 2026-08-08.
- [36] Josef Hofbauer and Karl Sigmund. 1998. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK. doi:10.1017/CBO9781139173179
- [37] Alireza Kavousi, Duc V. Le, Philipp Jovanovic, and George Danezis. 2026. Blind-Perm: Efficient MEV Mitigation with an Encrypted Mempool and Permutation. In *29th Int. Conf. on Principles of Distributed Systems (OPODIS 2025) (LIPIcs, Vol. 361)*. Schloss Dagstuhl, Dagstuhl, Germany, 36:1–36:21. Preprint: IACR

- ePrint 2023/1061. doi:10.4230/LIPIcs.OPODIS.2025.36
- [38] Mahimna Kelkar, Soubhik Deb, Sishan Long, Ari Juels, and Sreeram Kannan. 2023. Themis: Fast, Strong Order-Fairness in Byzantine Consensus. In *Proc. 2023 ACM SIGSAC Conf. on Computer and Communications Security (CCS '23)*. ACM, New York, NY, USA, 475–489. doi:10.1145/3576915.3616658
- [39] Mahimna Kelkar, Ertem Nusret Tas, Maryam Bahrani, and Tim Roughgarden. 2026. Designing Incentives for Responsive Consensus Protocols. IACR Cryptology ePrint Archive, Paper 2026/1234. <https://eprint.iacr.org/2026/1234>
- [40] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. 2020. Order-Fairness for Byzantine Consensus. In *Advances in Cryptology – CRYPTO 2020 (LNCS, Vol. 12172)*. Springer, Cham, Switzerland, 451–480. doi:10.1007/978-3-030-56877-1_16
- [41] Davide Mancino and Hasret Ozan Sevim. 2026. SoK: The Evolution of Maximal Extractable Value, From Miners to Cross-Chain. arXiv:2603.07716 [cs.CR].
- [42] Huneed Materwala, Shraddha M. Naik, Aya Taha, Tala Abdulrahman Abed, and Davor Svetinovic. 2025. Maximal Extractable Value in Decentralized Finance: Taxonomy, Detection, and Mitigation. *IEEE Trans. Services Computing* 18, 6 (2025), 4386–4407. doi:10.1109/TSC.2025.3620604
- [43] Jason Milionis, Ciamac C. Moallemi, and Tim Roughgarden. 2025. Automated Market Making and Arbitrage Profits in the Presence of Fees. In *Financial Cryptography and Data Security (FC 2024), Part I (LNCS, Vol. 14744)*, Jeremy Clark and Elaine Shi (Eds.). Springer, Cham, Switzerland, 159–171. doi:10.1007/978-3-031-78676-1_9
- [44] Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. 2022. Automated Market Making and Loss-Versus-Rebalancing. arXiv:2208.06046 [q-fin.MF].
- [45] Burak Öz, Benjamin Kraner, Nicolò Vallarano, Bingle Stegmann Kruger, Florian Matthes, and Claudio Juan Tessone. 2023. Time Moves Faster When There is Nothing You Anticipate: The Role of Time in MEV Rewards. In *Proc. 2023 Workshop on Decentralized Finance and Security (DeFi '23)*. ACM, New York, NY, USA, 1–8. doi:10.1145/3605768.3623563
- [46] Burak Öz, Danning Sui, Thomas Thiery, and Florian Matthes. 2024. Who Wins Ethereum Block Building Auctions and Why?. In *6th Conf. on Advances in Financial Technologies (AFT '24) (LIPIcs, Vol. 316)*. Schloss Dagstuhl, Dagstuhl, Germany, 22:1–22:25. doi:10.4230/LIPIcs.AFT.2024.22
- [47] Julien Piet, Vivek Nair, and Sanjay Subramanian. 2023. MEVade: An MEV-Resistant Blockchain Design. In *2023 IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC)*. IEEE, Piscataway, NJ, USA, 1–9. doi:10.1109/ICBC56567.2023.10174966
- [48] Radius. 2022. MEV-resistant ZK-Rollups with Practical VDE (PVDE). ethresearch forum post. <https://ethresearch.ch/t/mev-resistant-zk-rollups-with-practical-vde-pvde/12677>. Accessed: 2026-08-04.
- [49] Antoine Rondelet and Quintus Kilbourn. 2023. Mempool Privacy: An Economic Perspective. arXiv:2307.10878 [cs.CR].
- [50] Caspar Schwarz-Schilling, Fahad Saleh, Thomas Thiery, Jennifer Pan, Nihar Shah, and Barnabé Monnot. 2023. Time is Money: Strategic Timing Games in Proof-of-Stake Protocols. In *5th Conf. on Advances in Financial Technologies (AFT '23) (LIPIcs, Vol. 282)*. Schloss Dagstuhl, Dagstuhl, Germany, 30:1–30:17. doi:10.4230/LIPIcs.AFT.2023.30
- [51] Shutter Network. 2026. Shutter: A Threshold-Encrypted Mempool. <https://www.shutter.network/>. Deployed on Gnosis Chain. Accessed: 2026-06-12.
- [52] Patrick Spiesberger, Nils Henrik Beyer, and Hannes Hartenstein. 2026. Why Ethereum Needs Fairness Mechanisms that Do Not Depend on Participants' Altruism. arXiv:2603.05666 [cs.DC].
- [53] Pascal Stichler. 2025. Galaxy Era: Agent-based Simulation of Execution Tickets. arXiv:2501.16090 [cs.GT].
- [54] Aikaterini-Panagiota Stouka, Julian Ma, and Thomas Thiery. 2025. Multiple Proposer Transaction Fee Mechanism Design: Robust Incentives Against Censorship and Bribery. arXiv:2505.13751 [cs.GT].
- [55] Thomas Thiery, Francesco D'Amato, Julian Ma, Barnabé Monnot, Terence Tsao, Jacob Kaufmann, and Jihoon Song. 2024. EIP-7805: Fork-choice Enforced Inclusion Lists (FOCIL). Ethereum Improvement Proposals, no. 7805, status: Draft. <https://eips.ethereum.org/EIPS/eip-7805>. Accessed: 2026-06-12.
- [56] Mohammad Amin Vafadar and Majid Khabbazian. 2023. Condorcet Attack Against Fair Transaction Ordering. In *5th Conf. on Advances in Financial Technologies (AFT 2023) (LIPIcs, Vol. 282)*. Schloss Dagstuhl, Dagstuhl, Germany, 15:1–15:21. doi:10.4230/LIPIcs.AFT.2023.15
- [57] Toni Wahrstätter. 2026. MEV-Boost Dashboard. <https://mevboost.pics/>. Accessed: 2026-06-12.
- [58] Toni Wahrstätter, Matt Solomon, Ben DiFrancesco, and Vitalik Buterin. 2022. ERC-5564: Stealth Addresses. Ethereum Requests for Comment, no. 5564, status: Final. <https://eips.ethereum.org/EIPS/eip-5564>. Accessed: 2026-08-08.
- [59] Shan Wang, Ming Yang, Yu Liu, Yue Zhang, Shuaiqing Zhang, Zhen Ling, Jian-nong Cao, and Xinwen Fu. 2025. Time Tells All: Deanonimization of Blockchain RPC Users with Zero Transaction Fee. In *Proc. 2025 ACM SIGSAC Conf. on Computer and Communications Security (CCS '25)*. ACM, New York, NY, USA, 3490–3504. doi:10.1145/3719027.3765082
- [60] Yitian Wang, Yebo Feng, Yingjiu Li, and Jiahua Xu. 2026. Enshrined Proposer Builder Separation in the Presence of Maximal Extractable Value. arXiv:2601.12989 [cs.DC].
- [61] Fei Wu, Danning Sui, Thomas Thiery, and Mallesh Pai. 2025. Measuring CEX-DEX Extracted Value and Searcher Profitability: The Darkest of the MEV Dark Forest. In *7th Conf. on Advances in Financial Technologies (AFT '25) (LIPIcs, Vol. 354)*. Schloss Dagstuhl, Dagstuhl, Germany, 26:1–26:23. doi:10.4230/LIPIcs.AFT.2025.26
- [62] Sen Yang, Kartik Nayak, and Fan Zhang. 2025. Decentralization of Ethereum's Builder Market. In *2025 IEEE Symp. on Security and Privacy (SP)*. IEEE, Piscataway, NJ, USA, 1512–1530. doi:10.1109/SP61157.2025.00157
- [63] Sen Yang, Burak Öz, Fei Wu, and Fan Zhang. 2026. Geographical Centralization Resilience in Ethereum's Block-Building Paradigms. *Proc. ACM Meas. Anal. Comput. Syst.* 10, 2 (2026), 1–42. doi:10.1145/3805637
- [64] Sen Yang, Fan Zhang, Ken Huang, Xi Chen, Youwei Yang, and Feng Zhu. 2024. SoK: MEV Countermeasures. In *Proc. Workshop on Decentralized Finance and Security (DeFi '24)*. ACM, New York, NY, USA, 21–30. doi:10.1145/3689931.3694911
- [65] Haoqian Zhang, Louis-Henri Merino, Ziyang Qu, Mahsa Bastankhah, Vero Estrada-Galianes, and Bryan Ford. 2023. F3B: A Low-Overhead Blockchain Architecture with Per-Transaction Front-Running Protection. In *5th Conf. on Advances in Financial Technologies (AFT 2023) (LIPIcs, Vol. 282)*. Schloss Dagstuhl, Dagstuhl, Germany, 3:1–3:23. doi:10.4230/LIPIcs.AFT.2023.3