Thomas Thiery - February 12th, 2024

Thanks to Anders, Barnabé and Francesco for helpful comments and feedback on the draft.

tdlr: Just like in “The price is right”, predictive MEV-burn incentivises builders to predict the block’s full value at the observation deadline, without going over

[

87av1e (1)

890×499 143 KB

](https://ethresear.ch/uploads/default/original/2X/1/198693405fd270370ecbf9db9dc12fe724443435.jpeg)

MEV

Maximal Extractable Value

ePBS

enshrined Proposer-Builder Separation

EL

Execution Layer

CL

Consensus Layer

Introduction

MEV-burn is a consensus-level proposal to smooth and redistribute MEV rewards to ETH holders (i.e.,

). In its most recent iteration, MEV-burn was presented as an ePBS add-on to mitigate the security and economic instabilities caused by the fluctuating and unpredictable nature of MEV rewards. As a topic of active research, MEV-burn pros and cons are currently being examined from both theoretical and empirical standpoints. These efforts have sparked new ideas and lively discussions (see comments in the aforementioned ethresear.ch posts) around designing incentive-compatible mechanisms that prevent collusion between builders and proposers to ensure MEV is efficiently captured and burned, not unfairly diverted. This post aims to (1)

summarise vanilla and kickback MEV-burn key features and tradeoffs and (2)

introduce pMEV-burn, and propose research directions that might be worth exploring.

MEV-burn recap

MEV-burn as a concept is an attempt to mitigate negative externalities caused by MEV from a protocol perspective. The spiky, unpredictable nature of MEV profits generated by block proposers has led to two main issues:

1. Economic instability:

Proposers are rewarded not just from fulfilling their consensus duties but also from substantial execution layer MEV earnings. This dual revenue stream complicates the macroeconomic design of Ethereum (i.e., issuance curve), and leads to undesirable outcomes like “overpaying for security” and an increased validator count. At the microeconomic level, solo stakers also find themselves at a disadvantage compared to large staking pools, which have lower variance and are more likely to mutualise their profits.

1. Security threats

: Large MEV spikes also give incentives to proposers to execute all kind of ”rugpool”, reorgs, and DoS **attacks (see Justin and Mike posts for more details) that can impact the protocol’s security and potentially lead to catastrophic events.

To mitigate these issues, MEV-burn borrows its core properties from EIP-1559: First, the protocol must become MEV-aware, and gain knowledge about the “contention” value generated from the EL. Once captured, the protocol must redistribute that value fairly amongst network participants. One way to redistribute that value equally to all ETH

holders is to simply burn it. Notably, the recent Execution Tickets proposal also aims to curb the adverse effects of MEV, proposing to burn the entire cost of the tickets. This approach effectively removes MEV from staking revenue. However, we believe that further investigation into a live auction, single slot MEV-burn mechanism might still be warranted in this context.

Such a mechanism could potentially (h/t Francesco):

- Increase the amount burned, thereby reducing the portion of the burn that is discounted to offset the risk associated with buying execution tickets (as execution tickets tend to capture a smaller fraction of MEV).
- Enhance censorship resistance by factoring in base fees as part of the criteria for block inclusion, rather than solely relying on priority fees.
- Diminish the likelihood of multi-block MEV by imposing additional constraints on the winning block for each slot.

With that in mind, let's dive into vanilla MEV-burn (vMEV-burn) implementation details.

The step-by-step mechanism for vMEV-burn has already been described and illustrated elsewhere, but let's go through it once more so we can settle notations and use it to describe improvement proposals that build on top of it:

[

Screenshot 2024-01-26 at 08.53.14

1478×772 83.4 KB

](https://ethresear.ch/uploads/default/original/2X/d/d988df5b487a5d90ba22ba50c2f2f06916dfca00.png)

1. During slot n-1

, builders submit block headers alongside bids composed of (1)

the bid's base fee

b_{BF}

, which represents the amount of ETH

that a block will burn and (2)

the bid's tip

b_{tip}

paid to the proposer (also in ETH

).

1. N

seconds before slot n boundary SB

(usually N-2

), referred to as the observation deadline D

, validators on the attesting committee locally set a subjective Base Fee Floor (BFF

) based on the highest b_{BF}

they have observed so far.

1. At the beginning of slot n SB

(around N = 0)

, the proposer set its BFF

selects, signs and publishes the block header with the highest bid's tip, conditioned on the b_{BF}

being higher than their subjective BFF

.

1. The N

seconds between SB

and D

are meant to give attesters time to set their BFF

so that later on, they can vote for the proposer's block header if is valid, and matching its bid (i.e., the header chosen by the proposer).

1. If everything goes right, the selected block header starts gathering attestations and the builder releases the block payload containing the list of all transactions included in the block.
2. Attesters vote for the builder's payload if valid and matching its bid b_{BF}

.

1. If all is good, the b_{BF}

is burned and the b_{tip}

is paid to the proposer

vMEV-burn criticisms

vMEV-burn, by making sure the protocol can credibly capture and redistribute MEV, addresses both economic and security concerns. However, two aspects of the vMEV-burn design were challenged by astute observers (h/t jasalper, cometshock, ethdreamer and aelowsson)

and correctly identified in Mike's Dr. changestuff post:

- Collusion

: If MEV-burn is implemented, all the juicy MEV will be burned and redistributed to all ETH holders. Surely there must be a way for us, hardworking builders and proposers, to capture some (all?) of that value instead? Turns out there is.

- Builder-Builder

collusion**:** If all builders cooperate and make sure the base fee floor never gets set above 0 ETH

, builders can grief the proposer and share all the juicy MEV rewards between themselves. In the worst case scenario, all

builders would have to play along, as a single defecting builder setting a bid BFF

above 0 ETH

would be enough to break the scheme. That's a big if

but it's possible, since today 90.46%

of blocks are built by the top five block builders. Other more realistic scenarios involve a top cartel of builders, who collude to bid just enough to beat the best bid from other builders outside of their cartel, which could lead to a large delta even with a BFF

bid above 0 ETH

. Another argument I haven't seen too much is that the proposer could also choose to build a block locally, set the bid BFF

above 0 ETH

, and select its own block to make sure colluding builders have to outbid the BFF

and can't capture the whole MEV value. On this note, if participants were to behave honestly after ePBS and MEV-burn are implemented, [local block builders would have to (1) subsidise their own bids' BFF

](https://ethresear.ch/t/mev-burn-incentivizing-earlier-bidding-in-a-simple-design/17389/14) to remain competitive with other bids made by sophisticated builders or (2) decide not to build locally if the difference between their bid and the BFF

is too high. It's also important to note that while Builder-Builder

collusion is possible today, and somewhat orthogonal to MEV-burn itself, enshrining a mechanism allowing collusion between builders in the protocol is different than accepting the current out-of-protocol tradeoffs. Finally, Builder-Builder collusion would easily be spotted, and griefing proposers or forcing their hands, knowing they're the ones ultimately selecting the blocks that make it onchain, might not be strategically sound.

- Proposer-Builder

collusion: A similar, but slightly more elaborate collusion scheme involving all builders and the proposer can be set up. This time, colluding builders all agree to keep the bid BFF

at 0 ETH

and redirect the entire MEV value into the bid's proposer tip b_{tip}

(this can all be done via side channels as well). None of the value is burned by the protocol and redistributed to ETH

holders, instead the proposer can keep some of the value (e.g., equal to the b_{tip}

it would've collected under honest/normal conditions) and redistribute the rest of the value equally (or not) amongst colluding builders. This set up requires participation from more participants (the proposer has to be in on it too), but is potentially more sustainable and harder to detect if only part of the value is redirected to colluders via side channels. In the Builder-Builder case, builders are explicitly griefing proposers: this could lead to proposers just disconnecting from colluding builders and going back to local block building. In the Proposer-Builder case, both parties are involved, and play against the protocol. Another important point is that builders are not explicitly incentivised to bid before the observation deadline: A builder might want to place a bid right before the deadline, thus defecting from the collusion and winning one particular auction. But builders play in a repeated game, and in this case Proposer-Builder collusion is always worth doing for builders under quasi-perfect competition. Any value they can get away with by colluding will be higher than the close to zero profit margin they can make in a competitive landscape, given that the builder profit margin = actual value a builder can extract from its payload - (b_{BF}

(burned) + b_{tip}

).

- Builder-Builder

collusion**:** If all builders cooperate and make sure the base fee floor never gets set above 0 ETH

, builders can grief the proposer and share all the juicy MEV rewards between themselves. In the worst case scenario, all

builders would have to play along, as a single defecting builder setting a bid BFF

above 0 ETH

would be enough to break the scheme. That's a big if

but it's possible, since today 90.46%

of blocks are built by the top five block builders. Other more realistic scenarios involve a top cartel of builders, who collude to bid just enough to beat the best bid from other builders outside of their cartel, which could lead to a large delta even with a BFF

bid above 0 ETH

. Another argument I haven't seen too much is that the proposer could also choose to build a block locally, set the bid BFF

above 0 ETH

, and select its own block to make sure colluding builders have to outbid the BFF

and can't capture the whole MEV value. On this note, if participants were to behave honestly after ePBS and MEV-burn are implemented, [local block builders would have to (1) subsidise their own bids' BFF

](https://ethresear.ch/t/mev-burn-incentivizing-earlier-bidding-in-a-simple-design/17389/14) to remain competitive with other bids made by sophisticated builders or (2) decide not to build locally if the difference between their bid and the BFF

is too high. It's also important to note that while Builder-Builder

collusion is possible today, and somewhat orthogonal to MEV-burn itself, enshrining a mechanism allowing collusion between builders in the protocol is different than accepting the current out-of-protocol tradeoffs. Finally, Builder-Builder collusion would easily be spotted, and griefing proposers or forcing their hands, knowing they're the ones ultimately selecting the blocks that make it onchain, might not be strategically sound.

- Proposer-Builder

collusion: A similar, but slightly more elaborate collusion scheme involving all builders and the proposer can be set up. This time, colluding builders all agree to keep the bid BFF

at 0 ETH

and redirect the entire MEV value into the bid's proposer tip b_{tip}

(this can all be done via side channels as well). None of the value is burned by the protocol and redistributed to ETH

holders, instead the proposer can keep some of the value (e.g., equal to the b_{tip}

it would've collected under honest/normal conditions) and redistribute the rest of the value equally (or not) amongst colluding builders. This set up requires participation from more participants (the proposer has to be in on it too), but is potentially more sustainable and harder to detect if only part of the value is redirected to colluders via side channels. In the Builder-Builder case, builders are explicitly griefing proposers: this could lead to proposers just disconnecting from colluding builders and going back to local block building. In the Proposer-Builder case, both parties are involved, and play against the protocol. Another important point is that builders are not explicitly incentivised to bid before the observation deadline: A builder might want to place a bid right before the deadline, thus defecting from the collusion and winning one particular auction. But builders play in a repeated game, and in this case Proposer-Builder collusion is always worth doing for builders under quasi-perfect competition. Any value they can get away with by colluding will be higher than the close to zero profit margin they can make in a competitive landscape, given that the builder profit margin = actual value a builder can extract from its payload - (b_{BF}

(burned) + b_{tip}

).

- Late bidding

: A separate, but related issue around late bidding behaviour from builders has also beenraised. Even if builders don't intentionally collude and form a "late-bidding cartel" (h/t to Anders for the cool name) to keep the bid BFF

at 0 ETH

, it turns out that a lot of MEV coming from CEX-DEX arbitrages grows over time in expectation, and is almost always captured during the last second of the slot. This raises valid concerns: if MEV-burn only captures value until SB -2

, but significant value still goes to the proposer via the tip, we're back to square one, and the mechanism isn't efficient in doing what it's supposed to. While Toni and Mike both presented interesting estimations of how much value would be burned vs tipped to the proposer based on historical data, it's important to keep in mind that builders' bidding behaviour will also evolve based on the incentives they're presented with or external, unpredictable changes in MEV opportunities and market conditions.

- Observation deadline games

: In the vMEV-burn proposal, at observation deadline D

, validators on the attesting committee and the proposer locally sets their subjective BFF

based on the highest b_{BF}

they have observed so far. This means builders are expected to compete and raise their b_{BF}

up to that point. The expected behaviour is to see builders raise their b_{tip}

after D

has passed, up until the slot boundary SB

since we expect proposers to select the winning bid based on the highest tip, as long as the b_{BF}

is higher than the BFF

.

- However, since [there is no consensus on what the BFF

value is (it's set locally)](https://ethresear.ch/t/mev-burn-incentivizing-earlier-bidding-in-a-simple-design/17389/6), builders and proposers will have to make probabilistic judgements when they observe bids made very close to D

: "If I observe a bid at D + 0.5s with the highest b_{BF}

, maybe the majority of attesters are better connected than I am, observed it at D

, and will treat it as the BFF

so my own b_{BF}

has to match it?". This means sophisticated, well-connected proposers will be able to bid more competitively than solo stakers, since they have more confidence regarding whether the b_{BF}

they observed was seen by the rest of the network or not.

- An attacker could also cause consensus instabilities by bidding at D

with the highest b_{BF}

, and b_{tip} = 0

so that no other builders can outbid it, hoping that the unsophisticated proposer will ignore the attacking bid (since b_{tip} = 0

) and knowing that choosing any other bid will be invalid because it won't meet the BFF

requirements, thus causing an empty slot. Since there is no incentive for builders to bid before D

in the vMEV-burn design, the cost of this attack could potentially be very low if the equilibrium strategy ends up being waiting after D

to start bidding, and the BFF

set by other builders is kept around 0 ETH

.

- However, since [there is no consensus on what the BFF

value is (it's set locally)](https://ethresear.ch/t/mev-burn-incentivizing-earlier-bidding-in-a-simple-design/17389/6), builders and proposers will have to make probabilistic judgements when they observe bids made very close to D

: "If I observe a bid at D + 0.5s with the highest b_{BF}

, maybe the majority of attesters are better connected than I am, observed it at D

, and will treat it as the BFF

so my own b_{BF}

has to match it?". This means sophisticated, well-connected proposers will be able to bid more competitively than solo stakers, since they have more confidence regarding whether the b_{BF}

they observed was seen by the rest of the network or not.

- An attacker could also cause consensus instabilities by bidding at D

with the highest b_{BF}

, and b_{tip} = 0

so that no other builders can outbid it, hoping that the unsophisticated proposer will ignore the attacking bid (since b_{tip} = 0

) and knowing that choosing any other bid will be invalid because it won't meet the BFF

requirements, thus causing an empty slot. Since there is no incentive for builders to bid before D

in the vMEV-burn design, the cost of this attack could potentially be very low if the equilibrium strategy ends up being waiting after D

to start bidding, and the BFF

set by other builders is kept around 0 ETH

.

kickback MEV-burn

In his recent post, Anders proposed a new mechanism, referred to as kickback MEV-burn (kMEV-burn) to incentivise builders to bid earlier than the observation deadline D

. The changes are purely additive to the vMEV-burn design, and can be summarised as follows (focus on the purple writing

and the

in the figure below):

[

Screenshot 2023-11-23 at 13.10.13

1634×958 114 KB

](https://ethresear.ch/uploads/default/original/2X/b/b8fb3bb6a45b10145f5ed6728fc1f03d8ffebb20.png)

1. Same as in the vMEV-burn design, other steps that are strictly the same will be skipped
2. At step 2, attesters and the proposer not only set their subjective BFF

based on the highest b_{BF}

they have observed at D

, but also remember the winning builder's identity (i.e., its public key).

…

1. When attesters vote for the proposer's selected block if its b_{BF}

exceeds their BFF

, they also vote TRUE

in a separate vote if the winning builder's identity is the same as the one that submitted the block selected by the proposer. Otherwise they vote FALSE

.

…

1. In the end, if the majority of attesters in the slot vote TRUE

, the winning builder receives a kickback x

, which is a fraction of the winning block's payload base fee wb_{BF}

that otherwise would have been fully burned (e.g., x

= 0.05).

The builder rewards R_{builder}

are determined by the kickback x

can be expressed as:

R_{builder}= x * wb_{BF}

They come in addition to the profits (or loss) the builder has made from the difference between the actual value the builder has extracted from

wb_{BF}

and the (b_{BF}

(burned)

- b_{tip}

). The proposed kMEV-burn changes are meant to address "late-bidding cartels" forming between builders by explicitly rewarding honest builders that compete to win the auction and set an accurate BFF

before at the observation deadline D

. Another aspect worth considering is that observation deadline games can still be played under the kMEV-burn design: an attacking builder can still trick proposers into selecting invalid bids raising the BFF

right at D

, but if other builders are incentivised to win the auction until D

, the cost of the attacker would be significantly higher and might incur a loss if it ends up overbidding and being selected by the proposer.

kMEV-burn example s

cenarios

In this section, we hope to give an intuition on how builders could structure their bids at the observation deadline D

and at the slot boundary SB

under kMEV-burn by introducing example scenarios. The first one illustrates what a naive, conservative builder would do. We assume attesters' local BFF

is the same as the naive builder's b_{BF}

at D

, and that the block chosen by proposer is the one that was built by the same builder:

[

Screenshot 2023-11-29 at 11.25.19

1402×640 55.7 KB

](https://ethresear.ch/uploads/default/original/2X/f/ff1f77cb64713a251ee835733cb48e7682c2bcc8.png)

1. At D

, the naive builder observes a given block value, say of 1.2 ETH

, and decides to set his b_{BF}

to the full block's value to increase its chances to set the BFF

and be remembered by a majority of attesters.

…

1. After D

has passed, the builder decides to redirect the additional block's value to b_{tip}

, that will be rewarded to the proposer.

1. If b_{BF}

≥ BFF

and the block value is 1.4 ETH

at SB

, then b_{tip}

= 0.2 ETH

and b_{BF}

=1.2 ETH

. If x = 0.05

, assuming the block chosen by the proposer was built by the same builder who set the BFF

, rewards from the kickback mechanism can be computed as: 0.05 * 1.2 = 0.064 ETH

We can draw several observations from this first example:

- This example scenario assumes near-perfect competition: If a builder is convinced he has access to higher value than any other builder (e.g., because he has access to exclusive order flow), it will probably not bid its full value at D

, and would rather bid the lowest value that’s above what other builders will be able to bid, according to his estimates.

- Kickback rewards determined by x

have to be high enough so that builders are incentivised to use the mechanism, rather than keeping the difference between the block’s actual value and their b_{BF}

for themselves.

- After D

has passed, all the additional value captured in the last 2 seconds are redirected to the bid tip and goes to the proposer. This behaviour doesn’t seem to be rational, and we suggest sophisticated builders will either increase their b_{BF}

**to get a higher kickback, or keep the excess value to themselves rather than giving it to proposers via b_{tip}

.

Based on the last observation, we think builders will naturally tend to (1)

bid up to the expected full block’s value in their b_{BF}

at D

to make sure it sets the highest BFF

and (2)

return little to no value to the proposer in their b_{tip}

. This is an issue, because it means proposers won’t really have the option of selecting bids with the highest b_{tip}

. Instead, they’ll be forced to choose the one bid that satisfies the BFF

, even if that bid’s b_{tip} = 0

. One way to deal with this is to let proposer rewards go to 0 or close to 0. After all, we might want proposers to only get compensated by CL rewards. We can also imagine builders would iterate over very low tips, (e.g., submit block at D with BFF = 10

and b_{tip} = 0

, then send a block with BFF = 10

and b_{tip} = 0.00000001

, etc…) and keep going until the end, ensuring their b_{tip}

is high enough to lead and get selected by the proposer. Here’s an example of what it might look like if the builder is really good at predicting the future total MEV in a given slot:

[

Screenshot 2024-01-08 at 14.34.25

1508×582 49 KB

](https://ethresear.ch/uploads/default/original/2X/e/ec3c1303e2bcc5ebafe03843a031082e8db65006.png)

1. At D

, the predictive builder observes a given block value of 1.2 ETH

, and decides to set his b_{BF}

**based on its expected full block’s value at SB

. In this example say the builder expects a full block value of 1.4 ETH

, and bids b_{BF}

=1.4 ETH

and sets b_{tip}

= 0 ETH

.

…

1. Here, the predictive builder was perfectly accurate in its prediction. The full block's value at SB

is indeed 1.4 ETH

. Its b_{BF}

**was the highest at D

so a majority of attesters set their local BFF

to1.4 ETH

and remembered the builder's identity (voted TRUE

). In this case, b_{tip}

=0 ETH

, b_{BF}

=1.4 ETH

and kickback rewards are 0.05 * 1.4 = 0.07 ETH

. All good, except the proposer doesn't get any rewards from b_{tip}

!

kMEV-burn advantages

- Discourages "late-bidding cartels"

: Colluding builders setting a low BFF

will not be remembered by attesters, and will not receive builder rewards from the kMEV-burn mechanism.

- "Incentives to predict"

: The expected behaviour from competitive builders converging towards bidding the full MEV block value at D

is a feature, not a bug. It sets clear goals for builders, and emphasises competition over prediction capabilities that might give a small edge to builders that can build efficient prediction algorithms and low-latency transaction simulation, rather than pure order flow.

- Builders who are good predictoooors

can generate a guaranteed revenue stream even in near perfect competition and their margins trend towards zero.

kMEV-burn disadvantages

- Misaligned incentives

between builders and the proposer: There is a tension between betting the full expected block value in builders' BFF

at D

to win attester auction and giving tips to proposers. Under competitive settings, a plausible scenario would be proposer rewards going to or near zero. This situation could potentially encourage proposers to grief builders and opt for choosing their own payloads.

- Proposer extortion games

(h/t Barnabé): Growing the pie and adding builder kickback rewards can lead to situations in which the builder enters side-

channel agreements with the proposer to rebate some of the kickback rewards back to the proposer (e.g., “As a proposer, I will only select your block if you send half of your kickback rewards to me. Otherwise I’ll just select another competing bid and you’ll get nothing.”).

This type of side channel agreements was the main reason behind burning base fees in EIP-1559, and is a major concern since competing bids will likely have very similar values, so the proposer wouldn’t lose much by selecting another competing bid.

- Complexity

: kMEV-burn introduces a new task for attesters and as Anders suggested in his post, this can lead to games in which builders may seek to influence attestations.

- More attestation deadline games

: Since in kMEV-burn, attesters have to vote on the builder’s identity, attackers could send tons of bids with very small value differences, making it impossible for attesters to have an homogenous view of who the best builder (by BFF

) is at D

, and preventing kickback rewards to be distributed.

- Asking builders to predict the future block’s value might lead to more sophistication

, which is a known centralisation vector. However, one might argue we’re already way past trying to avoid too much sophistication from builders, and predicting how much value they can extract from MEV opportunities and efficient packing is exactly what competitive builders are doing today.

predictive MEV-burn

The predictive MEV-burn proposal builds on kMEV-burn’s design and is motivated by a first principles approach to builder incentives. If MEV-burn’s main objective is to capture and burn a block’s payload full value, then we should explicitly incentivise builders to accurately predict what the payload full value will be at the observation deadline D

(e.g., 2 seconds before the winning bid is selected at SB

). The proposal also aims to address some of kMEV-burn disadvantages by (1)

Simplifying how builders’ bids are structured by removing b_{tip}

and keeping b_{BF}

as a single variable to estimate the block value and (2)

realigning builder and proposer incentives by introducing:

- Builder rewards R_{builder}

set as a proportion x

of b_{BF}

set at D

, referred to as b^D_{BF}

or BFF

.

- Proposer rewards R_{proposer}

set as a proportion y

of the b_{BF}

at SB

, or b^B_{BF}

, for proposers

In this post, we assume x = y = 0.05

, but more research is needed to determine x

and y

absolute and relative values. If proposer/builder kickbacks are too high, then the whole purpose of implementing MEV-burn falls apart, as this will generate the same economic and security concerns as MEV today. If the kickback rewards are too low, then builders might not be incentivised to bid earlier than D

and might collude (amongst themselves or with the proposer) to keep the BFF

low and collect most of the MEV. In the pMEV-burn design:

- Builders

want to maximise their b_{BF}

at D

so they can set the BFF

, be seen and remembered by attesters and collect R_{builder}

.

- Proposers

want to maximise R_{proposer}

by choosing the block with the highest b_{BF}

at SB

.

- For a given slot, both the builder and the proposer only get rewarded of the majority of attesters voted TRUE

, meaning the block selected by the proposer was built by the builder that set the BFF

(see step 4. in figure below).

These changes mean builders and proposers incentives are now aligned, as there are no more incentives for proposers to extort builders for part of their kickback rewards (see Proposer extortion games

section in kMEV-burn disadvantages)

.

To mitigate attestations deadline games and level the playing field among attesters and the proposer, we also propose to introduce a short time buffer so builders stop bidding 300 ms before the observation deadline, as was suggested in the comments of Anders' post. This buffer increases the chances of participants having a more homogeneous view of the highest bid to set their local BFF

and remember the winning builder's identity. These changes are represented on the figure below.

[

Screenshot 2024-01-16 at 11.22.54

1720×884 129 KB

](https://ethresear.ch/uploads/default/original/2X/7/70be329c098823a8c7622167a0ec376f689f12ed.png)

We also describe and illustrate an example scenario for the happy case, in which a builder set the BFF

at D

to 1.4

. That builder was also selected by the proposer at SB

, and a majority of attesters voted TRUE

, meaning they had the same view as the proposer. In this case:

- BFF = 1.4
- R_{builder} = x * BFF = 0.05 * 1.4 = 0.07
- R_{proposer} = y * b^B_{BF} = 0.05 * 1.45 = 0.0725
- The rest of the block value is burned:

Burn = \text{Block value} - R_{builder} - R_{proposer} = 1.3075

[

Screenshot 2024-01-09 at 16.52.46

1594×512 60.7 KB

](https://ethresear.ch/uploads/default/original/2X/2/2e89d97e6fc2e53ee7f014e316ef9e02dd2df4a8.png)

In the in, the pMEV-burn proposal leaves us with a the following parameters to set:

- The attestation deadline D

time

- The time buffer before D

duration

- The proportion of BFF

given to the wining builder x

- The proportion of b^B_{BF}

given to the wining builder y

And more empirical/simulation-based analyses are needed to determine the initial values that should be given to these parameters.

pMEV-burn disadvantages

While we believe pMEV-burn presents some advantages outlined in the previous section, this design comes with its own limitations. Its main flaw can easily be illustrated if a large MEV spike happens after the observation deadline D

. In that case, the difference between BFF

and b^B_{BF}

is likely larger than the proportion of b^B_{BF}

that constitues R_{proposer}

in the pMEV-burn design. In these instances, we can imagine builders and proposers being willing to use side channels to share the value being generated by the MEV spike rather than including it in b^B_{BF}

and burning a large portion of it.

An important follow-up analysis to determine the viability of the pMEV-burn design would be to look at bids data submitted to relays and determine the percentage y

of total base fees b^B_{BF}

so that R_{proposer}

are greater than the amount of MEV between D

(e.g., at -2 seconds) and SB

. Results of this analyses might give good directions to set parameters, but it’s important to keep in mind that builders are not trying currently incentivised to maximise their bids at SB - 2

seconds.

Discussion and research directions

In this post, we set out to summarise key features and tradeoffs, map out the design space, and propose a new design for a single slot, live auction MEV-burn mechanism. Our goal is for this to contribute to the development of a robust and efficient mechanism that can both capture and alleviate the detrimental effects of MEV. While writing the post, we identified several research areas that require further development:

- Attester games: Understanding incentives for a majority of attesters to delay or change their votes (TRUE

or FALSE

), including potential use of side channels, bribes and extortion games. Explore possibilities of implementing other mechanisms beyond binary votes, like having a committee (similar to a Payload Timeliness committee) vote on the bid value and builder's identity directly (h/t Anders).

- Costs and profits from collusion: Simulations can help determine when Builder-Builder or Proposer-Builder collusion are profitable strategies compared to playing according to the mechanism's rules. In turn, this can help setting values for parameters like x

and y

for builder and proposer rewards, respectively.

- Auction design: MEV-burn designs described in this post assume an open, dynamic first-price auction similar to the MEV-boost auction. Other designs should be explored (e.g., sealed bid, to maximise the amount of MEV captured by the protocol and limit centralising forces that can emerge from latency games and strategic bidding behaviour.

This list is not exhaustive, and a key aspect of future research will be to explore methods of critically evaluating existing proposals, which could lead to the development of new and innovative ideas.