by @Mikerah, @sbaks0820, @yahgwai, @quintuskilbourn

, @parthgargava, @lsquaredleland

, @sxysun

during the IC3 Ethereum Summer camp

MEV-Boost is middleware to enable a market for MEV opportunities for Ethereum after the merge. In the current design, complete privacy is lacking and, as such, builders are required to trust relays not to act on privileged information. In this document, we aim to outline the various approaches to adding complete privacy to MEV-boost and provide some concluding thoughts on potential directions for research in this area.

Desiderata

An ideal complete privacy solution for MEV-Boost should have the following properties:

1. Simple dominant strategy for proposers
2. Permissionless
3. Trustless (parties cannot steal MEV from one another)
4. Low Latency

The first three properties are countermeasures against centralisation (arguably in descending order of importance) and the last property is a requirement of efficiency, both to maximise value extraction and to compete against other designs which may coexist with our proposed solution. No known, feasible design satisfies all desiderata at this at this time. The current design satisfies 1 & 4 and potentially satisfies 2, depending on the implementation of relay.

This document explores other designs which are ultimately different points in the space of tradeoffs between the desiderata.

Approaches

Builder release

[

490×503 9.2 KB

](https://ethresear.ch/uploads/default/original/2X/d/def5e065e1d424f5bed3642ecfe8a2ee7193952b.png)

The builder sends a header and bid to the relayer who forwards this information to the proposer. The proposer responds with a signed header, which is then forwarded back to the builder who then releases the block and signed header to the network. The builder can grief the proposer and other builders by bidding high enough to win and not releasing the block, causing the proposer to lose the block reward. This requires the relay to have a sophisticated reputation scheme to prevent empty slots from happening.

It's hard to prove misbehaviour of the builder, as it can release the valid block too late in the slot causing the proposer to miss the block reward. From the perspective of the proposer the relayer is responsible for any misbehaviour of the builder. In this scheme the builder is completely protected from having their MEV stolen.

The main weakness of this scheme is the low cost to builders to grief the proposer and cause empty slots. To improve this the relayer could require a bond from the builder. Upon broadcasting the block to the network, the builder would also be required to send the block to the relay which also broadcasts the block. The relay slashes the builder if the block came too long after the header was sent. Whilst this does mean that the builder needs to place trust in the relayer not to slash arbitrarily, the builder is protected from a any party stealing the contents of its block.

Proposer encrypted blocks

[

622×560 11.4 KB

](https://ethresear.ch/uploads/default/original/2X/2/2ef644b1e6f44ddc80f179ecb396796c2db18396.png)

The builder encrypts the block with the proposer's public key, so that when it's passed to the relayer the contents will not be revealed. The implication here is that the relayer cannot check the validity of the block before the getting a signature from the proposer. Like in the design above, builders now have a direct way to grief the proposer at very low cost, meaning that relayers will need much more sophisticated reputation schemes with the builder to ensure valid blocks. However, in this

design, it’s very easy for the proposer or relay to prove the builder was at fault, opening up avenues for trustless slashing via fraud proofs (see diagram below).

[

669×676 18.5 KB

](https://ethresear.ch/uploads/default/original/2X/3/38ef784053fb8fabd5e47f1a59c9fdc91e66b58a.png)

Given the desire to simplify the role of the proposer, one may wish to first rely on the relay to carry out the fraud proof and if this does not happen within a certain time period, the proposer is able to do it themselves and remove the relay from its whitelist.

An additional concern is that the proposer can collude with the relayer to decrypt the block and steal its contents, which is problematic (1) in that there is now another way for a sophisticated proposer earn more than its unsophisticated counterparts and (2) there is still a trust assumption involved for the builder, although this trust assumption is weaker than simply trusting the relay in the current model.

Note:

Shutter Network has a similar design (to be made public soon) which employs introduces a committee of key-holders that allow for threshold encryption to take place. This has the benefit of making proposer-relay collusion more difficult, at the expense of additional complexity. We leave discussion of that approach to a time when their proposal has been made public.

SGX

[

595×566 9.88 KB

](https://ethresear.ch/uploads/default/original/2X/2/2e53f945bc753b0f11e3fdbbfb3158a647bbf578.png)

This is the same as the current model with trusted relay, however, instead of relying on the relayer as a trusted third party, this role can be delegated to a secure enclave like SGX. With SGX, there are stronger guarantees that the relayer code hasn’t been tampered with which translates to stronger guarantees that blocks are valid and not stolen. In the case that the SGX is broken, we can default to the status quo in which a trusted third party (e.g. Flashbots) can run the relayer.

Builders still need to trust Intel (if SGX is the chosen enclave), but this trust assumption is much weaker than relying on the honesty of the relay operator. A further concern for the builder is that it is difficult to ensure that no untrusted party doesn’t have physical access to the machine in which the SGX enclave resides.

Finally, it is likely that the SGX operation will introduce some additional latency, reducing efficiency of extraction.

Note

: instead of waiting for cryptography to get better we can design sick applications now

.

MPC and ZKPs

[

593×669 14.3 KB

](https://ethresear.ch/uploads/default/original/2X/3/35a5b53d6fdd0caeccada97c05ef9ded26eb4db3.png)

Instead of relying on TEEs and economic incentives, it is possible to provide complete privacy to MEV-Boost by relying on cryptographic primitives, namely zero-knowledge proofs (ZKPs) and multiparty computation (MPC).

A builder provides a proof of that the encrypted blob they submitted is a valid block and a proof that their encrypted block pays the specified bid. The builder sends this to the relayer for verification. Once verified by the relayer, the relayer sends all the (header, bid) pairs to the proposer to select which bid they want. Once selected, the proposer sends to the relayer a signed header. Then, the relayer sends the encrypted block back to the proposer for decryption. The proposer can decrypt the block and propose it to the network. Significant increases of latency could be incurred by this design depending on how cryptographic tools evolve.

The role of the relay now becomes verifying the proofs which is presumably somewhat complex. The proposer still needs to trust the relay submitting blocks.

An alternative system could exclude the relay completely. This removes the need for the proposer to evaluate which relays

to trust, but likely introduces other, stronger centralising vectors like difference in proof verification rate and spam resistance.

Malicious actor(s)

MEV-Boost

Builder Release

Bonded fraud proof

ZKP

SGX

Relayer

Conceals block from proposer; Can steal MEV

Conceals block from proposer

Conceals block from proposer

Conceals block from proposer

None

Proposer

Can steal block if (MEV>cost of slashing)

None

Can steal block if (MEV>cost of slashing)

Can steal block if (MEV>cost of slashing)

Can steal block if (MEV>cost of slashing)

Builder

None

Can cause loss of block reward

Can accept slashing to cause empty slot by submitting invalid encrypted block

None

None

Relay + Proposer

None

None

Can steal MEV

None

None

Builder + Proposer

None

None

None

None

None

Builder + Relay

None

None

None

None

None

Threshold Encryption

Threshold encryption can be used instead of zero-knowledge proofs to provide complete privacy to MEV-Boost. A committee of proposers generate a shared public-private key pair in which each proposer has a share of the private key. Builders encrypt the transactions in their blocks with the shared public key. Some threshold of proposers would need to decrypt the blocks transaction and then include the block. The main drawbacks of this approach is that it introduces extra latency to block production and that joining a quorum of proposers may need to be restricted as most threshold encryption protocols introduce an honest majority assumption.

Witness Encryption

Witness Encryption is a technique that restricts decryption of a message to those that hold a witness to a condition that decrypts the message. With witness encryption, a proposer can only decrypt a transaction if and only if its hash has been included into a finalized block. Another approach using witness encryption that would be more in-line with PBS would be the have the proposer decrypt a block if and only if they can provide a valid signature to its block header. The main drawback of this approach is that at the time of writing there are no practical implementations of witness encryption.

Timelock Encryption

Timelock encryption allows one to decrypt a message once a certain amount of time has passed. In our context, builders would generate encrypted blocks and send them alongside a bid and an unencrypted block header to the proposer. The proposer chooses the most profitable bid and sends back a signed block header. Once a certain period of time has passed, then the block would be decrypted.

Conclusion

Although there is no clear optimal design, there is room for innovation around out-of-protocol PBS. Given the agnostic design of MEV-Boost, some of these designs can be implemented without any changes to MEV-Boost. This would add diversity and competition between the difference relays MEV-Boost clients are able to connect to. Future work further outlining and developing the cryptographic schemes required to enable some of the designs above, could prove to open the space of possible relay design even further.