Preface

- OH: “This was the least watched talk from Research Day, but possibly the most important one imo”
- Below you will find notes from @cwgoes presentation
- Talk

Introduction

The edge of MEV is a metaphor in the sense that its the edge of the current MEV discourse and that there is an interesting graphical representation of the whole question related to switching costs.

Four Quadrants of MEV

- Supply-side price fixing
- Delegation of computation
- Information flow asymmetry
- Cross-domain synchronization

Cross-Domain synchronization

Cross domain stuff is hard because of heterogeneous security and heterogeneous clocks. These are hard problems. See above.

Information Flow asymmetry

Most of what went under the term MEV is in this class

- Users have less information (unsophisticated agents)at the time of intent creation than operators (validators, builders, searchers) have at the time of ordering decision-making
- Order flow auctions, encrypted mempools with batch matching, etc. all try to address this by changing the information flow or adding auction mechanisms
- Most of the current “bad MEV” that we don’t want falls into this class.

Information asymmetry between the user at the time of intent submission and the orderer (~ proposer) at the time of ordering - as the latter is always after the former, the orderer will probably have access to more information, which they could use to reorder, insert their own intents/txs, etc. (this is the majority of current Ethereum MEV, as I understand it).

This can be addressed by quantizing time at a granularity large enough to span the difference between these two times (plus possibly some slack), which can be accomplished by threshold decryption (encrypted mempool), time lock/witness encryption, etc. Presently, the most practical technique here is threshold decryption (we have an implementation of that with Ferveo), but probably we’ll eventually want to move to witness encryption as it has better security properties.

Delegation of Computation

- User wants to delegate “hard work” (e.g. of optimal trade routing) to a third party
- Meta-transactions, specified builder algorithms, fairness criterion, etc.
- Most of the current “good MEV” falls into this class

Users wanting to outsource computation (and being willing to pay for it). This is similar to the category of “good MEV”, meta-transactions, this sort of thing - any case where users want someone else

to do some computation for them. Also includes potentially splitting trades between multiple domains (domains here could be multiple DEXs on one chain, multiple DEXs on different chains, speaking broadly).

Here our concern is to make sure that the market for compute is fair - compute is fungible, and users should get the best price - while also trying to avoid accidentally creating proof-of-work-style races which aren’t very efficient - but users clearly do want to outsource computation, so this is something we design for explicitly. In our organization of concepts, this also includes provable indexing, ETL, and the like.

Supply side price-fixing

This is simple, and maybe so simple that it's been neglected so far.

- Interdependent transactions (if we want them to touch shared state that can compose) must be ordered in one (logically central) location
- This location has some operator set
- That operator set chooses the price for inclusion

The interdependent location typically has an operator set, the fundamental reason why is that we don't want to make a liveness assumption of the users, If you want them to be only one class of participant than the users need to be online, but typically people don't like being online all the time, but we generally avoid this.

So we want to delegate liveness which delegates custody, delegating the ability to sign to do something with our assets on our behalf, to some operator set. That operator set fundamentally just by the fact of delegating liveness that operator set chooses what to include, as they choose what to include they set a price. You can try to create norms around what prices they set, but they always have the ability to choose what intent or transaction is included. They will always be able to accept side payments or set prices in some other way.

What is Supply-Side Price fixing?

Price fixing has a long-standing background in colloquial economic discourse (”Price fixing” implies a cartel, this is intentional, operators in consensus are a cartel)

Operators set prices

Its straightforward for operators to coordinate, we've built fancy BFT consensus mechanisms that literally make it easy for them to coordinate, they know the public keys and the voting powers. (Operators = validators and people who have custody of assets, supply side price fixing only has to do with whoever ultimately controls inclusion in the logically centralized ordering location)

”Price fixing” implies a cartel, this is intentional, operators in consensus are a cartel. True independent of supply/demand asymmetry (independent of all other types of MEV. Operators for a particular ordering location can set a price, they order things in a place, and they accept some type of payment for this);e.g., EIP 1559 is an algorithmic price-setting tool (a way operators coordinate to set a price), but operators still ultimately control.

Why Should You Care, though?

Supply-Side

- We should expect operators to coordinate
- Operators set prices for inclusion
- Operators are rational (ish) (they want to maximize profit depending on what they think will happen, repeated games and one shot games).
- Operators have custody

Demand-Side

- There are reasons to expect users need or want to agree on where they order things, where they order particular types of transactions.
- This agreement on one logical ordering location for interdependent transactions is what makes atomic composition possible.
- Even if you have something like Heterogeneous Paxos, you still need a logical single location for ordering.

Rational Extraction

Suppose that users don't coordinate, and they just send their transaction somewhere. Assume surplus from coordination. In a world where users don't coordinate, Operators can charge just under the surplus before it becomes individually rational for users to leave (this is a coordination problem). I as a user, if I benefit for using the system, but if the operator charges me $5 to send a tx

, even though it doesn't cost that to run the computers, they know they can charge $5, and I still send tx

, that's the individually rational equilibrium, users don't coordinate.

If operators expect users to leave, they should charge users as much as possible. This changes the game theory into a one shot game, if I am the validator set of the Ethereum blockchain and see everyone is taking their assets to other ordering locations, If I think people will stop sending me txs then the rational action for me is to charge as much as the assets are worth for the users to exit. If I as a coordinated set of operators expect users to leave, I should charge them as much as possible because this is the final round of the game. We don't see this yet, but we should expect to.

Another strike against the conceptual unity of "MEV"

- Supply-side price fixing is unrelated to information flow asymmetry
- TEEs, threshold encryption, OFA, etc, do not help (other than maybe make stuff harder for operators to measure).
- Operators can easily vary the fee to test how much they can extract

This category of MEV is unrelated as in you can do all the cryptography you want, you can do nice privacy preserving intent-centric architectures, OFAs, SUAVE, TEEs, it does not help with this problem. The only asterisks is that if users have heterogeneous security preferences such that they are willing to accept multiple ordering locations you can do some coordination amongst users in the counterparty discovery layer, in the intent layer but that only helps with better balancing transactions across a set of different logical ordering locations. This doesn't help with a bridge withdrawal tax or basic price fixing.

You may think these systems may make it harder for operators to figure out what users are doing. However, operators can still test different fee levels and see what users are willing to pay. If users don't coordinate, they will individually be willing to pay what using the system is worth to them.

Why don't we see this right now?

One kind of danger in the applied MEV research discourse is that we want to be careful that we don't build systems that we only see right now. Some of the problems won't happen until things are widely adopted.

For example, Ethereum validators are nice, there are side social channels of interdependency. If we look at the real-world outside the blockchain ecosystem, who extracts? The typical Silicon Valley business model has not only ended up exploiting switching costs, it's often explicitly designed to maximize switching costs. They are analogous to operators in this scenario. We can expect in the long-run, if operators see that people are leaving, they are going to have simple algorithms to recognize those transactions and charge higher fees.

What is the Slow Game?

- Users coordinating to select logical ordering location

The coordination process among users to select logical ordering locations. However, it is users agree where they want to send inter-dependent intents and transactions which need to be ordered together.

- Agreeing on where to send interdependent transactions/intents
- Splitting non-interdependent flows b/w operators

(This includes splitting up non-interdependent flows between operators so you avoid high prices due to high demand exceeding supply for one particular location if you didn't need interdependency.)

- Switching away from extractive operators by "force" (forking)

It also includes users need the ability to make a credible commitment that they will fork out extractive operators in a way that is sufficient to change the game theory.

You are Probably already playing the slow game.

The slow game is already happening, just socially rather than in a more algorithmic way.

- Using particular applications on particular chains (a kind of affinity for certain types of intents)
- Moving to rollups with lower fees (no exit tax price discrimination yet)
- Complaining that fees are too high and you will leave you are playing the slow game (making a threat not coordinated but a threat nonetheless)
- No, really
- No, really

Problems

- Messy, subject to propoganda
- Not very efficient
- High switching costs

The way it is currently played the slow game is messy and subject to propaganda, it’s not very efficient, especially with all of these heterogeneous protocols or websites, there are high switching costs.

A Modest Protocol Proposal

An interesting avenue of research. When faced with a coordination problem what you want to ritualize, often a viable solution is crafting both tech and culture for a protocol.

What do Operators Do?

- Custody Assets
- We make a liveness assumption of them
- Most popular: Ethereum validator set

If we reduce this problem to its essence here, all we are talking about in terms of operators is custody. Operators custody assets their signature is required, they can deny providing that signature, that’s why they can extract in this way. Then we make a liveness assumption of them

What do we as Users want?

- Users want to delegate to operators…but we don’t want the operators to charge “too much”
- i.e., users want to set prices
- Users (as a group) want to negotiate with operators (as a group)
- Users can win because they make the system valuable (i.e., operators want their tokens)

We as users want to delegate to operators. We don’t want to be online all the time, but we don’t want the operators to charge too much (hard to quantify). Furthermore, we want to set prices. If users coordinate, they have the ability to set prices because they bring value to the system because they choose to use it.

If we assume that most of these distributed systems are used as accounting techniques to track things going on in the external world. Users choose to create that correspondence between the external world and the accounting system by using it. Ultimately, they have the negotiating leverage. Operators want their tokens and users want to negotiate.

I think users could do this by periodically demonstrating the credible ability to coordinate and potentially agree or negotiate on prices to pay operators. To avoid this bridge exit tax problem, Users need to be able to make credible commitments that they will fork out any operators who attempt to extract value when they exit.

The Modest proposal

- We already have this great way to demonstrate the ability to coordinate…
- What if users just run consensus?

The good news is we don’t have to invent anything super complicated to solve this.

What must slow game consensus do?

- Demonstrate ability to coordinate
- Signal to other users which operators will be used
- Credibly threaten to fork out operators who try to extract
- Set price guidelines for operators (potentially)

Slow game consensus is the opposite thing of what you think of when you think of fast blockchains. Mostly we work on making blockchains faster; optimizing transaction processing, validators data center co-locations, etc.

Slow game consensus does not care about this. All we need to do is periodically demonstrate the ability to coordinate signal to other users to make it easier to coordinate on which operators will be used and credibly threaten when necessary to fork out operators who try to extract too much. Perhaps setting price guidelines will make things more efficient?

The Slow game is… Slow

Consensus protocols typically have at least N^2 message latency, running consensus amongst millions or billions of users won't happen in seconds, it will be days. We'll need a bunch of cryptography techniques like BLS-signature aggregation. We probably don't want a proposer just aggregate individual data. The slow game can be slow. It can be daily, monthly, less frequently. Users and operators typically are in a repeated game, operators want future revenue. So all users need to do is condition that future revenue on operators not setting prices too high. Then any operators who want to show up and claim that revenue, they can do so. If users set it too low, no operators will come. They can iterate on a negotiation until they get to a mutual agreeable level.

We don't discuss the particular economics, but the concept of a Solver DAO or Builder DAO is very relevant to this. If you expect operators to coordinate as a class, and perhaps there is some composition subgroup coordination within there, that class and users as a class are going to end up negotiating.

In particular, we also think it's helpful to have the ability to run the slow game on demand. Suppose that next week the Ethereum validator set decides, oh shit everyone is moving funds over bridges to rollups, and they are never going to come back. And they decide to set all of your bridge withdrawal prices to the value of ~95% of your tokens. So if you want to withdraw to a rollup they will say erc20.transfer you must give 95% to the validators, maybe socially distributed.

In order to deal with this, we need the ability to run this slow game on demand, or at least demonstrate that we can do so. Such that if validators try to do that, you can run the slow game and agree to copy the state of this particular location and change the operators. Ex. You would copy the entire state of the Ethereum blockchain (the accounting history you want to preserve) and set a new location with different operators. You need to do this atomically (you need to run consensus amongst the users)because this is a coordination problem. You don't want two forked Ethereum's competing. You want to switch all the state atomically to a new location.

Nothing New Under the Sun

- We're already running consensus
- Just not with a protocol
- A protocol (tech + culture) makes the threat more credible

In some sense, this is nothing new. We are already doing consensus. People are already talking amongst themselves about where they want to send transactions. But we don't have a protocol. A protocol would make the threat more credible and would provide a period test that we actually have sovereign networking infrastructure and the ability to fork out operators if they are too extractive. Luckily, this is not even that hard, we are just running consensus with a lot of people.

Network Topology

One way of thinking about things is that users are the disconnected dots or the large circle at the edge of the system. They interface with the system and the outside world. They ultimately control if they coordinate the constraints over what can happen inside the system. If users don't coordinate, the MEV gets sucked into the center. But if users coordinate, the center is paid some amount which is negotiated but effectively fixed by the users and everything keeps going.

End-state Equilibrium

- Users set fixed margin for operators
- Users want to set enough for secure setups, maybe not "brutal" competition
- Also references based on locality, trust, etc.
- Users run slow game periodically to keep things credible
- Prices go down slightly over time as compute gets cheaper
- Everyone is happy (?)
- Needs more match

One question which needs more mathematical formalization is what does the end-state equilibrium of this system looks like. There are a lot of dimensions of complexity and multi-variable preference optimization that we have completely glossed over and are not trivial, including; locality, low latency, trust. In general, we should aim towards a system where users coordinate. And in their coordination they set with several rounds of negotiation a fixed margin for operators. Users here were assuming have multiple variables they care about. They might not want race to the bottom margin competitions. Users may want resilience of operator sets. If you put everyone in a race to the bottom margin competition, to some extent you select against resilience.

Users may want to include more variables in how they select operators. But still, you would expect that prices will go down slowly over time as compute gets cheaper and prices get more standardized as the system becomes more streamlined. It seems to me that if you reach this equilibrium, everyone is roughly happy. Some operators are paid a constant cost (a multiple of networking, bandwidth, and hardware costs) users still have control over the system and no more than some finite amount of MEV is extracted by the operator class. But if we don't have a credible coordination game, even if we solve all the other MEV problems, it's a big deal. Especially once operators realize to some extent people are paying them because their assets are custodied not because they explicitly trust the Ethereum validator set. As protocols standardize, the market for operators becomes more competitive.

Conclusion and Takeaways

MEV is more than information asymmetry, overflow, compute delegation, or even cross-domain concerns (a fundamental part of MEV is just about delegation of custody. There is potential for extraction if users don't coordinate. We haven't seen this part of MEV yet in the world, but we probably will. Its a very dangerous kind because you don't see it until a system becomes embedded. Once a system is extremely embedded, that's where the extraction happens. If the switching costs are too high and users don't coordinate we see this with the real world, google has physical fiber lines, if your adversary winds up controlling the physical network infra you are S.O.L. You want to design against this before it happens accidentally.

- Users decide what is valuable, but to prevent the operators to whom they delegate agency from extracting that value, they must coordinate, and coordination requires consensus (social or otherwise)

- ”Slow game” consensus doesn't need to be fast, it just needs to be credible. A credible commitment changes the equilibrium.