Problem:

Reducing the impact of MEV often relies on “repeated game” quasi-reputational assumptions, trusted third parties, or ex ante specification of the form that MEV will take.

Possible Solution:

SCAM Mechanism

The SCAM is a subgame perfect mechanism for reducing MEV in one shot games among anonymous participants, at least one of whom is a human. For the purposes of explanation I’m going to assume a Vickrey auction (e.g. to elicit builder MEV value.) This isn’t essential to the mechanism but it’s easier to explain this way.

So in the classic second-price sealed bid auction, agent i

bids b_i

for a good and, if they win, they pay the second highest bid, \max_{j \setminus i} b_j

. Agent i

plays the dominant strategy, setting their bid b_i

equal to their valuation of the good, v_i

, and getting payoff:

\pi_i = \begin{cases} v_i - \max_{j \setminus i} b_j & \text{if } b_i > \max_{j \setminus i} b_j\ 0 & otherwise \end{cases}

Now add a twist. Upon winning, agent i

must pay the full v_i

into escrow. Other agents {j \setminus i}

can pay some positive amount s_j

to burn some amount of the v_i - \max_{j \setminus i}b_j

in escrow. Any amount s_j

is burned as well. For simplicity assume that both piles are burned in equal amounts, so i

's expected payoff is now

\pi_i = v_i - \max_{j \setminus i} b_j - E[s_j]

But because s_j

is burned, any positive s_j

results in a negative payoff for j

. So under standard rationality assumptions, s_j=0

. Anything else reduces payoffs and thus has a profitable deviation to 0. Thus with rational expectations we have E[s_j]=0

, and this is the standard second price auction again. This is the bot version, let’s say, where MEV is maximized.

Adding Rabin Utility Function

But now let’s consider that agent j

is a human that has a Rabin utility function:

U_j=\pi_j(c_j,c_i)+\alpha_j \pi_i(c_i,c_j)

From left to right, this says that j

's utility is an function of their payoff, which is a function of the choices c

made by i

and j

, as well as i

's payoff as a result of their choices. \pi

is positive but \alpha

can be positive or negative and thus determines how the payoffs of other agents enter the utility function.

Let \alpha_j=-1

if j

believes i

has been malicious. Now imagine that Agent i

won the auction because he's a ruthless MEV extracting SOB. Poor Agent j

had spent weeks white-hatting a smart contract exploit to help someone unlock funds when along comes ruthless Agent i

who front runs the transaction and keeps the funds for themselves.

Which is to say that \alpha_j=-1

, bigtime, and so Agent j

finds it thrilling to be able to set some of i

's money on fire. As a result of this a positive s_j

is subgame perfect.

Now we have a different game, in some interesting ways. Consider two block builders, toxic and benign. Toxic can extract more MEV and would normally win the bid, but they are afraid of being punished. Meanwhile benign, who is taking less MEV, values the block a little less but doesn't have to fear punishment. A high enough expected punishment means benign can win the auction over toxic

, fair and square. All one shot, subgame perfect, etc.

Some cool things about this mechanism:

- It induces a fairness equilibrium by targeting the right kind of MEV–the kind that causes human disutility due to unfairness. The Rabin Utility function is a fundamentally human one, and the subgame perfection of the SCAM comes from the pleasure we may feel when justice is done. Bots are free to do what they like.
- It does not require specifying the form MEV will take ex ante

, but only recognizing it ex post

.

- It can also add come collusion resistance to Vickrey. Second price auctions are vulnerable to a type of collusion where the first price winner can pay some small fee to the second highest bidder to not bid, thus ensuring them a lower price. Under the SCAM Vickrey, this just increases the available punishment surface. Thus, in theory, they could punished for any suspected collusion as well.
- ETH gets burned "sound money, bat signal" etc. But all in the name of fairness.

Thanks @barnabe for some thoughts and advice on this weird mechanism (all errors mine ofc)