Order but Not Execute in Order
Transaction Execution Reordering in Atomic Broadcast
Tiantian Gong, Aniket Kate

1. Motivation

Problem of Front-running in blockchains
- In general blockchain systems, validators consistently order and execute transactions submitted by users.
- In blockchains like decentralizes exchanges (DEXes), validators profit from inserting, censoring, and reordering transactions in a proposed transaction batch, called front-running.

Existing solutions
- Extend the security definitions of traditional Byzantine fault-tolerant (BFT) state machine replication (SMR) (or atomic broadcast (ABC)) problem to include batch order fairness and have order-fair ABC protocols[1,2,3] (Traditional BFT SMR/ABC problem studies maintaining a consistent transaction log among replicas in a distributed system. But traditionally, security requires safety and liveness but does not concern the explicit order of transactions or whether the proposer has inserted or deleted certain transaction(s))
- Proposer Builder Separation (PBS) / Tax the front-runners by charging priority fees[7]
- Blind transactions through encryption[4,5,6], e.g., threshold encryption and delay encryption.
- Blinding does not eliminate MEV opportunities because the contents of transactions may be inferred, and front-running can be implemented in its traditional form, e.g., act before observing the explicit future actions of a victim splitting whale orders into small volumes.
- Proposer Builder Separation (PBS) / Tax the front-runners by charging priority fees[7]

: General order fairness is impossible (Condorcet paradox).
- Blind transactions through encryption[4,5,6], e.g., threshold encryption and delay encryption.

: Blinding does not eliminate MEV opportunities because the contents of transactions may be inferred, and front-running can be implemented in its traditional form, e.g., act before observing the explicit future actions of a victim splitting whale orders into small volumes.

: This does not solve front-running but leaves the market to reach equilibrium through actions carried out by interdependent players in an ever-changing environment.

2. Contribution

- Combing frequent batch auction (FBA) and or-ABC as a defense against front-running for DEX.
- Compare welfare loss in or-ABC under two common market designs, FBA and Continuous limit order book (CLOB), to support FBA as a market design response.

3. FBA VS CLOB

- CLOB executes ordered transactions one by one
- FBA match orders in a batch according to price, and all matched counterparties settle trades at the same price

4. Welfare comparison

- Solution concepts: Markov Perfect Equilibria (MPE) for CLOB and a weaker notion, Order Book Equilibrium (OBE) for FBA (since stationary MPE may not exist).
- In equilibrium, FBA imposes less welfare loss if

(1) public information ($\lambda_{pb}$) concerning an asset’s fundamental value changes is released more often compared with private information ($\lambda_{pr}$). Because first, under FBA, market-makers have time to respond to public information and do not need to mark up the price. Second, under CLOB, an arbitrageur can front-run the liquidity providers in case of both public and private information releases, the market-maker then demands more markups in equilibrium to counter the risk.

(2) the batch auction frequency ($I$) is compatible with the arrival rates of different types of trading parties.

(3) smaller priority fees for submitting transactions into the blockchain system also increase the markups under CLOB. Because front-running market-makers is more profitable, resulting in the liquidity providers charging higher markups.

Fig 1. Example regions where FBA has less welfare loss, truncated at $\lambda_{pr} = \lambda_{pb} = 3$. An interactive graph for tuning parameters can be found here[8].

References