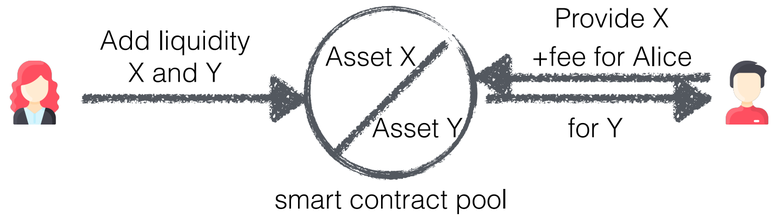
High-Frequency Trading on Decentralized On-Chain Exchanges

Liyi Zhou *, Kaihua Qin *, Christof Ferreira Torres †, Duc V Le ‡ and Arthur Gervais *
* Imperial College London, United Kingdom Email: liyi.zhou@imperial.ac.uk, kaihua.qin@imperial.ac.uk, a.gervais@imperial.ac.uk
† University of Luxembourg, Luxembourg Email: christof.torres@uni.lu
‡ Purdue University, United States Email: le52@purdue.edu

AMM DEX

- **Blockchains** enable peers to transact without trusting third-party intermediaries.
- **Smart contracts** are programs stored on the blockchain.
- **Decentralized exchange (DEXs)** allow parties to participate in financial markets while retaining full custody of their funds.
- **Liquidity Provider:** a market participant that provides liquidity.
- **Liquidity Taker:** a market participant that buys or sells one asset in exchange for another asset, by taking the liquidity offered by liquidity provider
- **Automated market maker (AMM) DEXs** algorithmically perform market making using smart contracts.

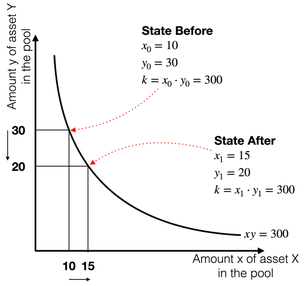


Constant Product Pricing Formula

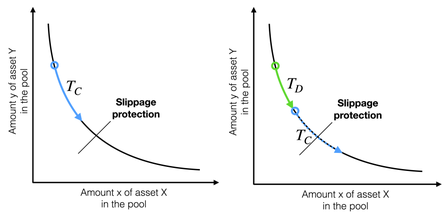
- Instant liquidity irrespective of the trade size
- Purchase of Y increases price of Y and decreases the price of X
- Ratio of asset X and Y sets the price

$$x \times y = k$$

asset X quantity | asset Y quantity | constant



Slippage Protection



There are two types of slippages:

- **Expected slippage** is the expected increase or decrease in price based on the (i) pricing formula; (ii) trading volume; (iii) available liquidity.
- **Unexpected slippage** is the additional slippage. This is typically caused by other market participants

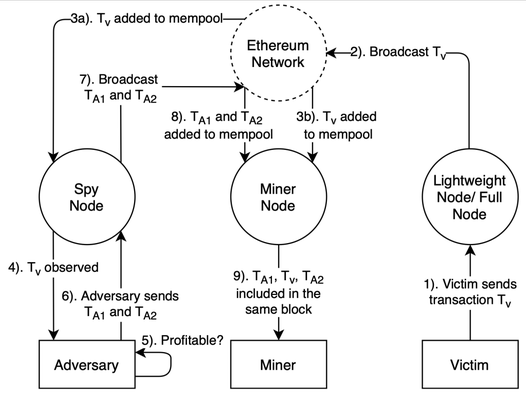
Predatory trading

In traditional financial markets, the predatory trading strategy of front-running involves exploiting non-public information about a pending trade. If the asset price is expected to rise/ fall as a result of the pending trade, the front runner will seek to buy/sell the asset before the pending transaction executes.

AMM DEXs aim to mitigate malpractice by providing complete transparency about (i) the available liquidity for asset X and Y; (ii) all performed trades; (iii) all pending trades on the P2P network; (iv) the pricing formula.

However, AMM DEXs also exacerbate malpractices, such as sandwich attacks.

Transparency + High-Frequency Trading = Attacks



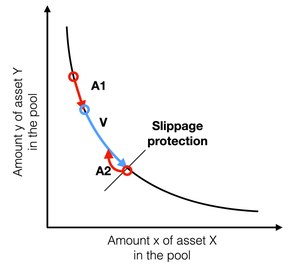
- We consider a blockchain P2P network, where a victim initiates trades on an AMM DEX.
- The adversary observes not yet mined pending victim transactions on the P2P network
- The adversary (not colluding with a miner) can issue its own transactions.
- The adversary manipulates the transaction "priority" by controlling the transaction fee per unit of computation

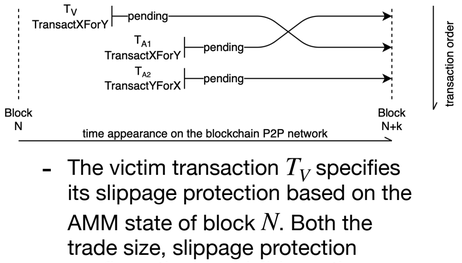
How miners order transactions

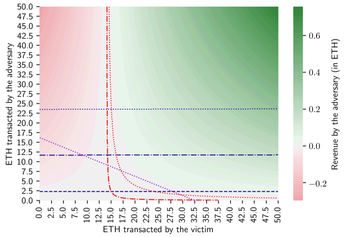
Strategy	Number of Blocks	Ratio
Empty Block	55,545	0.0234
Order per Gas Price	1,862,800	0.7853
Order per Parity Default	384,150	0.1620
Unknown Ordering	69,589	0.0293
Total	2,372,084	1.0000

- At the time of writing this paper, 78.3% of the Ethereum clients operate Geth, 20.2% operates Parity.
- Miner seems to switch strategies, but most blocks just sort transactions by their gas price per unit of computation at the time of writing (block 6.62~9M)
- Parity priorities local and retracted transactions first, and penalise transactions with heavy computation.
- Transaction ordering is more complicated nowadays, as miners start to provide transaction reordering as a service.

Sandwich Attack - Liquidity Taker Attacks Liquidity Taker

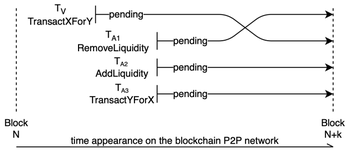






- The victim transaction T_V specifies its slippage protection based on the AMM state of block N. Both the trade size, slippage protection configuration are visible on the blockchain P2P network.
- The adversarial's goal is to include T_{A1}, T_V, T_{A2} in the same block. $N + k$ in that exact sequence.
- Not every victim transaction yields a profitable attack. ***[Mitigation]*** We quantify a minimum profitable victim input, under which an adversary will be unable to make a profit.

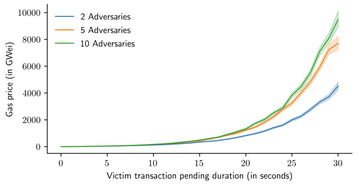
Sandwich Attack - Liquidity Provider Attacks Liquidity Taker



- We present a novel sandwich attack, where a liquidity provider targets a victim liquidity taker.
- The attacker pays higher transaction fees
- The attacker foregoes the commission fees for the victim's transaction

Multiple adversaries

- We assume all adversaries are rational and attack with the parameters defined in table below.
- Our results suggest that having multiple attackers does in expectation divide the total revenue.



Transaction Execution Order (left to right)	Winner	Reward for Attacker A	Reward for Attacker O
T_{A1} ✓ T_{O1} ✗ T_V ✓ T_{O2} ✗ T_{A2} ✓	A	Revenue − Fee(T_{A1}) − Fee(T_{A2})	−Fee(T_{O1}) − Fee(T_{O2})
T_{A1} ✓ T_{O1} ✗ T_V ✓ T_{A2} ✓ T_{O2} ✗	A	Revenue − Fee(T_{A1}) − Fee(T_{A2})	−Fee(T_{O1}) − Fee(T_{O2})
T_{O1} ✓ T_{A1} ✗ T_V ✓ T_{A2} ✗ T_{O2} ✓	O	−Fee(T_{A1}) − Fee(T_{A2})	Revenue − Fee(T_{O1}) − Fee(T_{O2})
T_{O1} ✓ T_{A1} ✗ T_V ✓ T_{O2} ✓ T_{A2} ✗	O	−Fee(T_{A1}) − Fee(T_{A2})	Revenue − Fee(T_{O1}) − Fee(T_{O2})