Battle of the Bots: Flash loans, Miner Extractable Value and Efficient Settlement

March 16, 2024
Preliminary

Abstract

Settlement on decentralized ledgers is transparent and batched. The settlement also allows settlement agents to expropriate profitable arbitrage trades. Arbitrage may be socially beneficial or wasteful. We model the effect of an alternate, private settlement on arbitrage. We document payments from arbitrageurs to private settlers that exceed 2 million USD per day.

Keywords: Blockchain, Decentralized Finance, Miner Extractable Value

Battle of the Bots: Flash loans, Miner Extractable Value and Efficient Settlement

Abstract

Settlement on decentralized ledgers is transparent and batched. The settlement also allows settlement agents to expropriate profitable arbitrage trades. Arbitrage may be socially beneficial or wasteful. We model the effect of an alternate, private settlement on arbitrage. We document payments from arbitrageurs to private settlers that exceed 2 million USD per day.

1 Introduction

Arbitrage activity is an integral part of modern financial markets. Arbitrageurs ensure that prices are consistent across different trading venues and that such prices are not stale. Profit maximizing arbitrageurs tradeoff the cost of capital for each leg against potential risky profit from trading divergent prices. The tension between this cost and benefit determines the limits to arbitrage and hence the efficiency of prices.

In decentralized finance, arbitrageurs play an even more important role. In addition to equalizing prices across trading places, they monitor collateral, and test the resiliency of smart contracts. Different from modern financial markets, capital for arbitrageurs in these markets is very cheap but arbitrage opportunities can be more risky. These differences arise from differences in the novel design of clearing and settlement in decentralized finance. In this paper, empirically and theoretically, we investigate arbitrageurs' incentives to find and exploit profitable arbitrage opportunities and investigate how these incentives change with changes in clearing and settlement.

Decentralized ledgers are a new type of settlement system that differ from traditional markets in two ways. First, trades are ordered in a batch and second, orders are exposed before they are settled. The fact that orders are batched means that any agent can propose a sequence of trades that are conditioned on each other. This credible commitment has led to unique order types such as flash loans which enable anyone to initiate arbitrage trades. Thus, decentralized ledgers make arbitrage easier. The fact that orders are exposed means that settlement agents (miners) or other observers can trade ahead of profitable trading opportunities submitted for settlement, and so free ride on the efforts of arbitrageurs who originate these opportunities. These trades decrease incentives to find profitable arbitrage opportunities. Cumulatively, do these features lead to socially efficient levels of arbitrage?

We present a simple model that illustrates the tradeoffs between efficient and inefficient arbitrage. In as much as arbitrage leads to more efficient prices and safer smart contracts, a social planner encourages this activity but is indifferent to transfers between agents. However, miners who appropriate any arbitrage opportunities effectively inhibit it. In such a world, a private market for settlement may increase welfare. Our model provides insights into when private settlement increases welfare, and the tradeoffs faced by arbitrageurs, settlers and ordinary users of the system. To interpret our model, we present stylized facts on flash loans (the cheap arbitrage capital) and more broadly arbitrage activity. We also document private trades between miners and arbitrageurs.

Our data allows us to distinguish between arbitrageurs that perform socially useful trades, and those that appropriate others' trades. We identify one group as *good bots* as they perform socially desirable tasks such as contributing to price discovery or maintaining systemic stability. DeFi lending platforms rely on members of the general public, so called keepers, to enforce the liquidation of under-collateralized loans. Keepers track the loans that the platform has issued and compare the market value of the collateral with the outstanding loan amount. If they correctly identify an under-collateralized loan they initiate a transaction with the lending platform, repay the outstanding loan, and obtain the collateral at a discounted price. The timely liquidation of loans by keepers are essential to maintain the financial stability of the platform.

While the existence of arbitrage opportunities is seen as a sign of inefficiency in traditional markets, arbitrage between decentralized exchanges (DEX) is part of their design. Decentralized exchanges are automated market makers that allow users to buy and sell tokens against an inventory. The automated market maker (AMM) is a piece of computer code on a blockchain and is therefore uninformed with respect to the current market price of a token. The AMM relies on arbitrageurs to move its price back to the market price. Arbitrageurs thus ensure that decentralized exchanges offer competitive prices. Other agents we refer to as *bad bots* engage in socially undesirable activities such as front running traders on decentralized exchanges.

Both kind of arbitrageurs have to invest a non-trivial amount of effort into identifying these trading opportunities. Liquidators of loans have to extract data from the blockchain, collect market data, and have to keep up with frequent changes in lending protocols' inner workings to identify under-collateralized loans. DEX-arbitrageurs have to quickly cycle through millions of possible trading paths between tens of thousands of liquidity pools to identify arbitrage opportunities. This task requires significant computing power and the development of sophisticated algorithms that need to be continuously updated as new DEXs or new pools on existing DEXs get deployed.

If these arbitrageurs deploy their transactions through the regular transaction channel they risk being exploited by other bots or by miners. In its original design Ethereum transactions get submitted to a peer to peer network and all transactions that await processing are kept in the publicly visible mempool. Sophisticated users could analyze the pending transactions and front run the arbitrageur by submitting the same trades with themselves as beneficiaries and a higher fee so that the miner would execute their transaction first. The miner, however, is in the best position to front run everyone. She has ultimate control over which transactions get included in the block and in which order they are executed. She can easily copy all profitable trades from the mempool and execute her transactions before those of the arbitrageurs and anyone who tried to front run the arbitrageurs. The value that miner could obtain this way is often referred to as Miner Extractable Value (MEV).

Such a an outcome, however, is not very likely. If the miner extracts all the value then arbitrageurs have no incentive to invest any effort in discovering trading opportunities which would endanger the DeFi system as prices are misaligned and undercollateralized loans do not get liquidated. In practice most miners run nodes that allow arbitrageurs to submit bundles of transactions directly to a specific miner for a fixed fee. We label these transactions that bypass the mempool and go directly to miners as private. Private transactions are not publicly visible until they are mined and can therefore not be front run by other bots. As we show below arbitrageurs often split the gain they make from a transaction with the miners.

We present several stylized facts that are consistent with miners extracting value from users of DeFi platforms. We see a dramatic risk in blocks in which transactions not executed in the order of the highest fee. Rational miners should prioritize transactions that offer higher fees per cost of execution, commonly referred to as the gas price. We find a steady increase in blocks for which transactions are ordered differently and on some days in May and June 2021 more than 80% of mined blocks contain transactions that are not prioritized on gas prices. The rise in unusually ordered blocks coincides with the wide adoption of MEV-GETH, a fork of the most popular Ethereum node which explicitly allows users to submit private transactions directly to miners.

Most private transactions are done by bots. We provide anecdotal evidence of private transactions and document how profits are shared between the arbitrageur and the miner. We trace one specific bot who conducted over 3,000 private transactions and paid over USD 2 million to miners. We then classify bots into good and bad bots based on whether their activity is socially desirable. Using a conservative approach to identify private transactions we find that good bots are more active than bad bots. After Mach 2021 bots transfer on average over two million USD per day to miners with 70.97% coming from good bots.

We document how settlement in an unregulated competitive market for settlement affects price discovery, information production, and the stability of the financial system. Our research is beneficial for regulators as it provides a base case how unregulated competitive settlement works and what mechanisms arise endogenously to mitigate frictions and conflicts of interest.

The term Miner Extractable Value was coined in Daian, Goldfeder, Kell, Li, Zhao, Bentov, Breidenbach, and Juels (2020), who classify ways in which miners could use their position for financial gain and analyze the implications of MEV for blockchain consensus. Several other papers in the computer science literature quantify some aspects of MeV. Qin, Zhou, and Gervais (2021) quantify MEV for specific protocols and selected transaction types. The estimate a MEV of 540 million USD over 32 months. Overall MEV is impossible to quantify because nobody could potentially evaluate all possible profitable transactions at a given time given state of the blockchain. Zhou, Qin, Cully, Livshits, and Gervais (2021) implement a novel search algorithm and showcase the computational complexity of finding profit taking opportunities from decentralized exchanges in 25 assets.

Capponi, Jia, and Wang (2021) present the tradeoff between a public mempool and a private market as the choice betwen Lit and Dark markets. Specifically, miners choose which venue to use (either one or the other). If few miners choose the dark venue then there is execution risk on the arbitrageurs. They find that aggregate welfare is highest if all miners adopt the dark venue.

2 Model

Consider a market in which settlement is performed by one miner who faces a cost of processing transactions normalized to zero, and charges a fee $f \geq 0$ for doing so. Transactions are generated either by agents who have a private value for transactions, or arbitrageurs whose transactions have a common value. A proportion ω of the common value trades also have a social value.

Every period, two private value customers are drawn whose valuation per transaction is $v > 0$, where $v \sim U[0, \bar{v}]$. A private value trader who does not consummate a transaction withdraws from the market. With probability λ strategic agents are drawn. There are two types of strategic agents: arbitrageurs who trade for profit, and screeners.

An arbitrageur can exert costly effort, $c_a(e) = \frac{ae_a^2}{2}$, to generate a common value trade of size $R > 0$ with probability e_a. With complementary probability, the arbitrageur generates a private value trade of $\bar{v}$. A screener can exert private effort e_s at a private cost $c_s(e) = \frac{se_s^2}{2}$. Exerting effort allows the screener to identify the common value trade with probability e_s, which he can then expropriate.

At most two transactions can be processed per period. For simplicity we assume that strategic agents are processed preferentially. We assume that the common value trade is sufficiently large so that $\omega R > \bar{v}$. There is no discounting and all agents are risk neutral. The sequence of moves conditional on an arbitrageuer being drawn is illustrated in Figure 1 below.

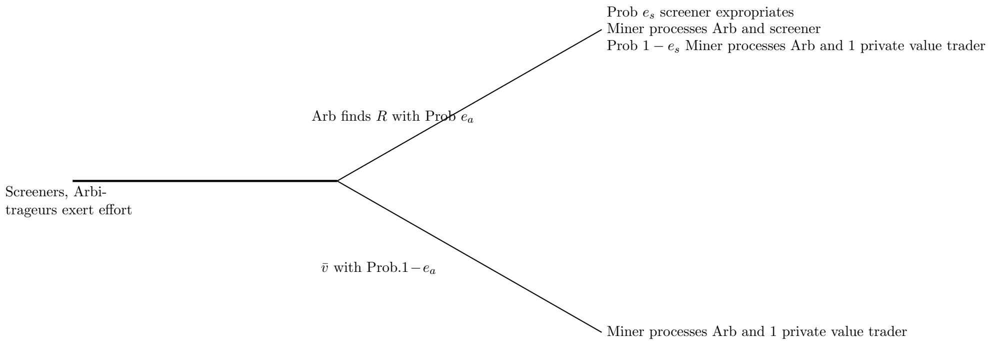
Prob e_s screener expropriates
Miner processes Arb and screener
Prob $1-e_s$ Miner processes Arb and 1 private value trader
Arb finds R with Prob e_a
Screeners, Arbi-trageurs exert effort
$\bar{v}$ with Prob.$1-e_a$
Miner processes Arb and 1 private value trader

Figure 1. Sequence of moves if an arbitrageur is drawn. Investments in screening and finding arbitrage are made ex ante. If an arbitrageur finds a common value trade and is expropriated, the miner processes both the arbitrageur and screener trades and no private value trades.

2.1 First Best

As we indicated in the introduction, some common value trades in the DeFi ecosystem also have a social value. Examples of such social value include liquidations in lending protocols that ensure risk free loans, cross-market arbitrage to ensure prices on-chain are aligned, and trades against smart contract vulnerabilities that lead to more robust code.[1] Let $0 < \omega < 1$ denote the fraction of common value trades that have social value.

The social planner has two slots to fill. As two private value trades arrive each period, they can always fill one slot with a private value trade. With probability λ, a strategic agent arrives, which either generates a common value trade or a large private value trade, both of which are valued by the social planner. The social planner is not concerned with transfers between screeners and the arbitrageur, but is concerned with the total common value trades that also have a social value. The planner's problem is therefore

$$\max_{e_a, e_s} E\Omega \quad = \quad (1 + (1-\lambda)) \int_0^{\bar{v}} \frac{v}{\bar{v}} dv + \lambda e_a \omega R + \lambda (1 - e_a) \bar{v} - \frac{a e_a^2}{2}. \tag{1}$$

[1] While trades against smart contract vulnerabilities are often described as "hacks" they perform the useful social function of identifying weak code. The scale and cost of such hacks should be evaluated relative to the scale and cost of regulatory rules, organizations and fines.

It is is immediate that the first best level of arbitrageur effort is $e_a^{fb} = \max\left[0,\ \lambda\frac{\omega R-\bar{v}}{a}\right]$. The first best level of screener effort is $e_s^{fb} = 0$. This is because the actions of the screener appropriates value that the arbitrageur has already found and is thus privately beneficial but not socially. We note that the welfare of the private value users is simply $(2-\lambda)\frac{\bar{v}}{2}$.

2.2 Nash equilibrium

Now consider the outcome when arbitrageurs and settlers interact strategically. An arbitrageur entering the market takes into account the expropriation risk when he decides to search for arbitrage opportunities. The arbitrageur's problem is to

$$\max_{e_a} E\pi_a = \lambda\left\{e_a R(1-e_s) + (1-e_a)\bar{v} - f\right\} - \frac{ae_a^2}{2}. \tag{2}$$

The problem of the screener is

$$\max_{e_s} \pi_s = \lambda e_s e_a\left(R-f\right) - \frac{se_s^2}{2}. \tag{3}$$

We summarize the best responses in the following lemma

Lemma 1 *The arbitrageur optimally puts in effort $e_a = \lambda\frac{R(1-e_s)-\bar{v}}{a}$, which is decreasing in screening e_s, while a screener optimally puts in effort $e_s = \lambda\frac{e_a(R-f)}{s}$ which is increasing in the arbitrageur's effort, e_a.*

The screener's actions extract profits from the arbitrageurs. This naturally reduces the effort that the latter are willing to put in to find profitable trades which affects the social surplus. Notice also that fees affect the marginal profitability of screening activity but not that of arbitrage activity.

Proposition 1 *In a public market in which miners are not screeners*

i. Arbitrageurs put in optimal effort $e_a^* = \lambda s\frac{R-\bar{v}}{as+\lambda^2 R(R-f)}$.

ii. Screeners put in effort $e_s^* = \lambda^2\frac{(R-f)(R-\bar{v})}{as+\lambda^2 R(R-f)}$

Both the arbitrageur and screener efforts depend on the equilibrium fees, which are chosen by the miner. Under the maintained assumption that if an arbitrageur arrives at the market, the common value is sufficiently high to warrant trade, the profit of a miner who is not a screener is

$$\begin{aligned} \pi_m &= f\left\{\lambda(1+e_ae_s)+(2-\lambda(1+e_ae_s))\frac{\bar{v}-f}{\bar{v}}\right\} && (4)\\ &= f\left\{2\frac{\bar{v}-f}{\bar{v}}+\lambda(1+e_ae_s)\frac{f}{\bar{v}}\right\} && (5)\end{aligned}$$

If the miner only faces liquidity trades, then the profit is simply $2f\left[\frac{\bar{v}-f}{\bar{v}}\right]$ and the miner faces the standard monopolist tradeoff: a higher price may deter trade. The optimal fee is $\frac{\bar{v}}{2}$. By contrast, strategic agentz are less price sensitive than the liquidity traders and always submit an order. Further, if the arbitrageur is screened, both they and the screener both submit transactions. In this case, there is congestion in the market and the private value traders are completely excluded.

Faced with arbitrage and screening activity, a profit maximizing miner will choose an optimal fee that maximizes Equation 5. This fee is implicitly defined by

$$2-4\frac{f_m}{\bar{v}}+2\frac{f_m}{\bar{v}}\lambda(1+e_a^*e_s^*)+\lambda\frac{f_m^2}{\bar{v}}\frac{de_a^*e_s^*}{df} \quad = \quad 0. \qquad (6)$$

We illustrate the optimal fee chosen by a monopolist miner in Figure 2. The monopolist miner's profit from private value transactions is maximized at $\bar{v}/2$. By contrast, the miner's profit increases linearly with arbitrageurs because they always submit a transaction, either because they have found an arbitrage opportunity or a private value transaction. If the arbitrageur is screened, the miner will also add the screening transaction.

Now suppose that the miner is also a screener. In this case, the miner receives additional profit from the screener, but does not pay the fee for the transaction.

$$\begin{aligned} \pi_{ms} &= f\Big\{\lambda+(2-\lambda(1+e_ae_s))\frac{\bar{v}-f}{\bar{v}}\Big\}+\lambda e_{ms}e_aR-\frac{se_{ms}^2}{2}. && (7)\\ &= f\Big\{\lambda(1+e_ae_{ms})+(2-\lambda(1+e_ae_s))\frac{\bar{v}-f}{\bar{v}}\Big\}+\lambda e_{ms}e_a(R-f)-\frac{se_{ms}^2}{2} && (8)\\ &= f\left\{2\frac{\bar{v}-f}{\bar{v}}+\lambda(1+e_ae_{ms})\frac{f}{\bar{v}}\right\}+\lambda e_{ms}e_a(R-f)-\frac{se_{ms}^2}{2} && (9)\end{aligned}$$

The difference in profit for the miner who is also a screener is that if the arbitrageur finds a profitable trade, the miner can expropriate it albeit without fee revenue for the transaction. This reduces the profitability of screening to the miner. Clearly, for a fixed fee, the optimal screening effort exerted by the miner screener is

$$e_{ms} \quad = \quad \lambda e_a\frac{\frac{f^2}{\bar{v}}+(R-f)}{s}. \qquad (10)$$

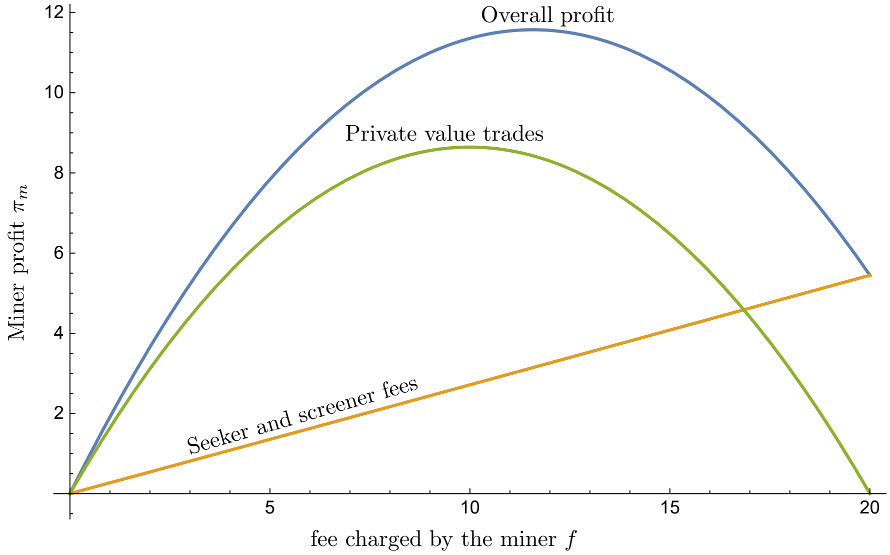

Figure 2. Profit for a miner who is not a screener as a function of the fee level. Profits from private value traders (green), from arbitrageurs and screeners (orange) and overall (blue). Parameters for the plot are $a = 7, s = 7, R = 40, \bar{v} = 20, \lambda = 0.25, \omega = 0.9$

On comparison with Lemma 1, for a fixed fee and arbitrageur effort, the miner screener exerts more screening effort than a standalone screener.

Proposition 2 *In a competitive market, with only public settlement in which miners are screeners*

i. Arbitrageurs put in optimal effort $\hat{e}_a = \lambda s \frac{R-\bar{v}}{as+\lambda^2 R[(R-f)+\frac{f^2}{\bar{v}}]}$.

ii. Screeners put in effort $e^*_{ms} = \lambda^2 \frac{(R-f+\frac{f^2}{\bar{v}})(R-\bar{v})}{as+\lambda^2 R[(R-f)+\frac{f^2}{\bar{v}}]}$

It is immediate that, for the same fee, there is less arbitrage activity in the public market in which miners are also screeners. However, we note that under the assumption that arbitrage payoffs are sufficiently large, the miner benefits from more arbitrage activity. As evident from the optimal effort exhibited in proposition 2, the lever with which to do this is through lower fees.

Proposition 3 *If the miner is also a screener then compared to the case in which the miner is not a screener,*

i. The optimal fee charged by the miner is lower.

ii. The expected welfare of the private value traders is higher.

iii. Arbitrage activity is lower.

We stress that our argument is that, conditional on there being screeners, welfare is higher if the miners are the ones who are doing it. This is because it provides them with an incentive to encourage arbitrage trades by reducing fees.

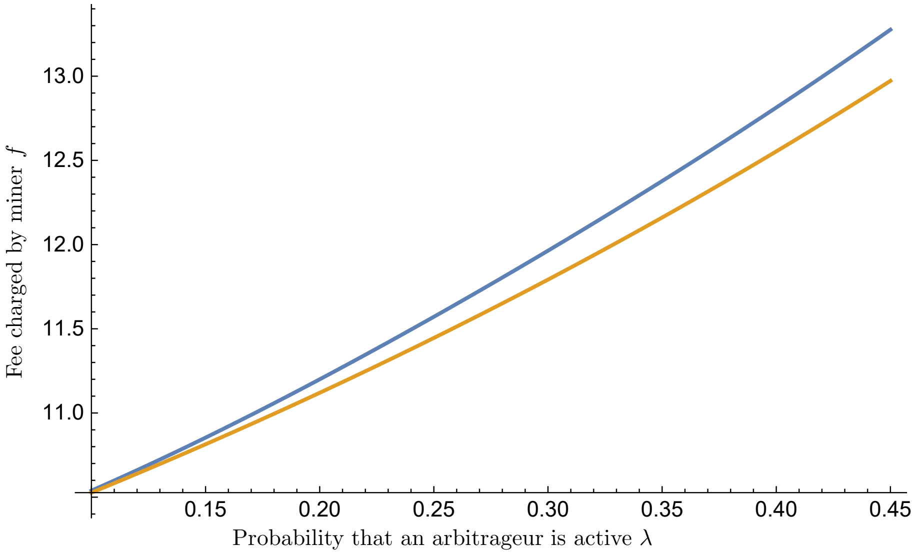

Figure 3. Optimal fee charges by miner who is a screener (orange) and who is not a screener (blue). Parameters for the plot are $a = 7, s = 7, R = 40, \bar{v} = 20, \omega = 0.9$

Figure 3 shows the optimal fee charges by miners who are screeners and those who are not as a function of an arbitrageur being active in the market. Miners who do not screen can only use the fee as a lever to extract surplus from the arbitrageur. They sacrifice fee revenue from private value traders in order to extract more from arbitrageurs and screeners whenever arbitrageurs are more active in the market (higher λ). By contrast, when miners are screeners they optimally keep fees lower to generate both higher volume from liquidity traders and more arbitrage activity from which to extract more value from the arbitrageurs.

3 Stylized Facts on Flash Loans and Arbitrage Efficiency

A natural question is to determine the size of arbitrage activity or a proxy for λ. While the entirety of arbitrage trades are impossible to identify, we use flash loans as a minimum measure of arbitrage activity. In this section we present stylized facts on flash loans.

One Ethereum transaction can interact with several smart contracts and call functions of these smart contracts to trigger economic actions such as borrowing, lending, conversion between tokens using a decentralized exchange, or transferring tokens between wallets. In a flash loan a borrower takes a loan at the beginning of a transaction and repays the loan at the end of the same transaction, thus repaying the loan at the same time as it was borrowed. Blockchain transactions are atomic, meaning that they either get executed in their entirety or not at all. Flash lenders face no credit risk.

Flash Loans were were invented in July 2018 by Marble, an open source lending platform on the Ethereum blockchain and combine the lending of funds.[2] Flash loans have experienced rapid growth with loans worth on average 1.17 billion USD borrowed per day in the first quarter of 2021 compared to USD 500,000 for the same period a year earlier.

The most common use-case for flash loans is arbitrage. Decentralized exchanges, which trade tokens worth billions of dollars each day, purposely rely on arbitrageurs to keep prices aligned with markets and consistent with each other. Flash loans provide cheap capital to arbitrageurs to execute their trading strategies. Other use cases for flash loans include swapping collateral for secured loans, loan liquidations, and exploits of weaknesses in other DeFi protocols. (A detailed description of their use gleaned from the computer science literature appears in the appendix.)

3.1 Sample

We collect flash loans from three leading providers: dYdX, Aave and Uniswap.

dYdX is a margin trading and lending platform. The platform is a popular source of flash loans which are available for a very low fee of 2 Wei, or 2×10^{-18} ETH. We observe 26,549 flash loans from dYdX in our sample issued on three tokens: wrapped Ether (WETH) and two USD stablecoins, USDC and DAI.

Aave is an open-source lending platform that has offered flash loans since January 2020. In January 2020 the protocol was upgraded to V2 and the old and new versions run in parallel. We collect 15,596 and 3,432 flash loans on V1 and V2, respectively for a 25 different tokens. Aave charges a fee of 0.09% of the flash loan amount.

Uniswap is a token trading platform. Uniswap consists of a family of liquidity pools, each consisting of two tokens that can be exchanged for each other. Flash loans in Uniswap are unique because a user can borrow an arbitrary combination of the two tokens and repay in a different combination as long as both have the same value. Over ten-thousand Uniswap liquidity pools exist, allowing users to borrow more tokens than other protocols. Uniswap charges a fee of 0.3% of the loan amount. We observe 5,841 flash loans from Uniswap in 381 tokens. For the most part of the analysis we focus on the 92.19% of flash loans that are against either WETH or one of three USD stablecoins (USDT, USDC, DAI). We end up with a total of 51,418 flash loans between December 16, 2019 and March 6, 2021.

Table 1 shows summary statistics of Flash loans per platform. The highest number of loans is on dYdX, which also has the largest loans, consistent with the lowest fees. Uniswap offers the greatest variety of tokens for flash loans and the relative high fee also includes a token trade in

[2] Marble was never widely used and is insignificant today.

Protocol	Mean Loan size	Median	Maximum	Number Loans	Number of Tokens
Uniswap	381,916	157	114,644,749	5,841	381
dYdX	3,411,871	71,854	272,122,064	26,549	3
Aave	206,821	3,607	183,296,205	19,028	25
Whole Sample	1,881,597	10,625	272,122,064	51,418	395

Table 1. Summary statistics on flash loans in USD per platform.

the liquidity pool where the loan was borrowed from making it making it an ideal platform for arbitrageurs. The average loan size in our sample is about 2 million USD, however, many loans are small. The median is USD 10,625 and 25.7% of loans are below USD 1,000. Few loans are very large with 5.6% of the loans in our sample for amounts larger than 1 million USD. The largest loan in our sample is for 151,332.4 ETH (approximately 272 million USD) on February 22, 2021 for what seems to be a triangular arbitrage transaction between Aave, Bancor, and the 1inch Exchange.[3]

Token	Mean Loan size	Median	Maximum	Number Loans	Volume
WETH	3,746,428	18,753	272,122,064	23,695	88,771,604,227
DAI	278,822	11,544	114,644,749	13,605	3,793,377,459
USDC	426,181	30,007	50,126,424	7,208	3,071,915,707
USDT	1,911,191	12,872	50,287,010	541	1,033,954,218
WBTC	43,220	11,790	1,140,086	606	26,191,105
LINK	50,719	5,569	8,307,375	240	12,172,619
BUSD	119,290	15,608	1,620,314	70	8,350,275
TUSD	59,248	10,643	1,534,530	114	6,754,297
YFI	27,120	2,928	660,303	138	3,742,497
YANG	5,880	393	135,164	422	2,481,155

Table 2. Summary statistics flash loans in USD per token for the 10 tokens with the highest aggregate lending volume.

We present summary statistics by token in Table 2 for the ten token with the highest aggregate lending volume. Wrapped ETH (WETH) is by far the most popular token to borrow in part due to its versatility. The most liquid liquidity pools in decentralized exchanges are trading some token against WETH. Next are the most popular USD stablecoins DAI, USDC, and USDT, followed by wrapped Bitcoin. Interestingly some less well known tokens also make the top 10 list. Flash loans for these tokens are only provided on Uniswap and loan sizes are small.

Figure 4 shows the daily volume of flash loans in USD. We can see that the volume of flash loans increases steadily over the sample period. 51.9% of the loans and 93.6% of the volume originate from dydx. Flash loan volume varies significantly over time with a clear growth trend. The average daily loan volume in January and February 2021 is 1,22 Billion USD, on average 247 loans are taken out and on average fees of 37,683 USD are paid per day by borrowers. Given the large volume of trades on Ethereum, this suggests a very low lower bound.

[3] see transaction 0x65781a5a076cece642bbd55cedf07c0bed379eda1314d25b8bea1b03a7176503

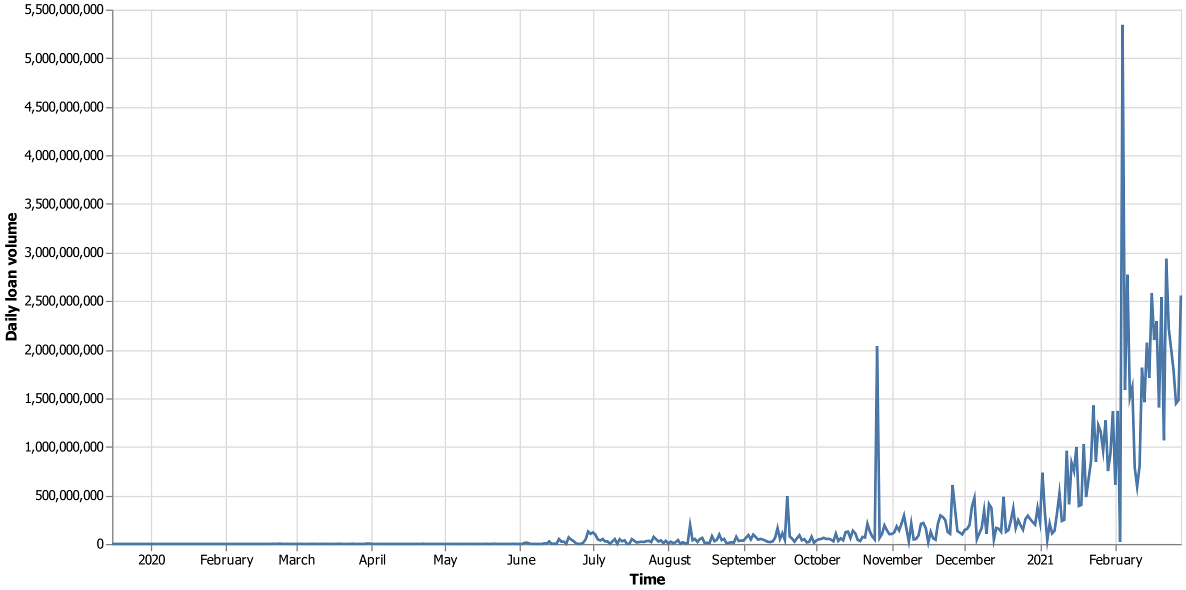

Figure 4. Daily volume of Flashloans in USD.

3.2 Private Settlement

Now suppose that there is a private market for settlement. In particular, the private market allows miners and arbitrageurs to bargain over the common value trades. For simplicity, we assume that if arbitrageurs are indifferent between the private and public markets, the arbitrageur will choose the private market. The sequence of events is illustrated in Figure 5 below.

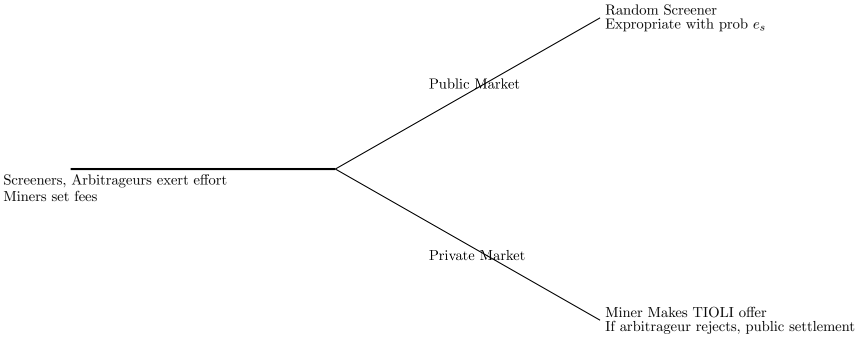
Random Screener
Expropriate with prob e_s
Public Market
Screeners, Arbitrageurs exert effort
Miners set fees
Private Market
Miner Makes TIOLI offer
If arbitrageur rejects, public settlement

Figure 5. Sequence of events with a private market if the arbitrageur finds a trade

First, consider the case in which the miner is not a screener. Let $\widetilde{e_s}$ denote the level of screening in the public market. If the arbitrageur goes to the public market, his payoff is $R[1 - \widetilde{e_s}] - f$. This payoff is his outside option. Thus, if a miner makes an offer x to the arbitrageur, in order for the arbitrageur to accept it, it must be that

$$\begin{aligned} R - x &\geq (1 - \widetilde{e_s})R - f \\ f + \widetilde{e_s}R &\geq x. \end{aligned} \tag{11}$$

A miner is willing to participate in the private market if it is better than his expected payoff in the public market or,

$$x \geq f \tag{12}$$

These are respectively, the highest and lowest feasible transfer to the miner. Given these bounds, for any offer $\hat{x}$, where $\hat{x} \in [f, f + \widetilde{e_s}R]$ both the arbitrageur and miner weakly prefer the private market.

Suppose the miner makes an offer $\hat{x}$. For this offer, the arbitrageur strictly prefers the private market and will accept it. If there is always an offer that the miner can make that the arbitrageur will accept, no common value trades will be processed in the public market. There is no incentive for the screeners to screen, as the only public transactions are private value ones.

Proposition 4 *Suppose that there is a private market, and no miners are screeners. Then, if the miner makes a take-it-or-leave it offer to the arbitrageur:*

i.) All arbitrage trades go through the private market and only liquidity trades are observed in the public market.

ii.) In equilibrium there is no screening and the private fee is equal to the fee in the public market $x = f$.

iii.) The arbitrageur optimally exerts $e_a^{private} = \lambda \frac{R - \frac{\bar{v}}{2(1-\lambda)}}{a}$.

If all trades go through the same market as in Subsection 2.2, screeners expropriate part of the value found by the arbitrageurs. By contrast, using a private market is a way for the miners and arbitrageurs to split this surplus between themselves. However, the threat of screening allows the miner to extract a large portion of the arbitrageur's profits. If the latter do not go to the public market, the screeners will not operate there which reduces the amount that the miner can extract from the arbitrageur.

There are two equilibrium effects of the private market: first, it increases the fees paid by private value traders and second, it eliminates screening. This increases the incentives of arbitrageurs to find profitable trading opportunities. If any of the arbitrage opportunities are beneficial ($\omega > 0$) this will increase social welfare. The net effect of the increase in fees to private value traders and the potentially effect of an increase in beneficial arbitrage is ambiguous.

Now, consider the polar opposite case in which the miner screens. In this case, there is maximal miner extractable value or MEV. As before, in this case the maximum amount that an arbitrageur would be wiling to pay for private settlement is given by Equation 11. However, in this case the minimum amount that is profitable for the miner is

$$x \quad \geq \quad f + \widetilde{e_s}R. \tag{13}$$

This higher reservation amount for the miner reflects the fact that if he also screens, his outside option includes any MEV from the public market. Thus, the only transfer that is consistent with the private market is $x = f + \widetilde{e_s}R$.

In this case, the miner-screener's ex ante profit becomes:

$$\lambda\Big(e_a(f + \widetilde{e_s}R) + (1 - e_a)f\Big) + (2 - \lambda)f\frac{\bar{v} - f}{\bar{v}} - s\frac{e_s^2}{2} \tag{14}$$

which implies an optimal screening level of $e_s = \frac{\lambda e_a R}{s}$.

The arbitrageur's profit is

$$\lambda\left(e_a(R - f - e_sR) + (1 - e_a)(\bar{v} - f)\right) - a\frac{e_a^2}{2}, \tag{15}$$

which implies an optimal effort level of $e_a = \lambda\frac{R(1-e_s)-\bar{v}}{a}$. We obtain

Proposition 5 *Suppose that the miner screens and there is a private market. Then, in the private market, if the miner makes a TIOLI offer to an arbitrageur, the*

i. The arbitrageur optimally puts in effort $e_a = \frac{s\lambda(R-\bar{v})}{as+(\lambda R))^2}$

ii. The miner screeners put in effort $e_s = \frac{R(R-\bar{v})\lambda^2}{as+(\lambda R)^2}$

iii. The miner charges a fee $f = \frac{\bar{v}}{2-\lambda}$.

The existence of the two markets allows the miner to price discriminate. The threat of screening in the public market ensures that they can extract a high fee from the arbitrageur in the private market. Because they only settle private value trade in the public market, they can post a lower fee to capture a larger market share.

Proposition 6 *Compared to a standalone public market, the existence of a private market for arbitrage trades*

i. Leads to lower fees and higher volume of liquidity trade in the public market.

ii. Leads to higher welfare for liquidity traders

iii. Leads to less congestion the public blockchain.

This is consistent with the original MEV Flash bots white paper, that suggested a private settlement market would reduce congestion on the blockchain.[4] In our environment this occurs because the miner screeners never "screen" in the sense of submitting orders, but rather trade in tandem with the arbitrageurs. Through the fee channel we highlight, liquidity traders also benefit from the private market. We note however, that the investment in screening technology which is required to extract rents from the arbitrageurs is a socially inefficient investment.

With the change in proof of work to proof of stake, the Ethereum system can move to one in which block builders are separate from miners. This so called proposer-builder separation allows more specialization. Specifically, proposers will order transactions and auction them to builders who validate the blocks.[5]

In the context of our model, the miner both chooses "builds" and proposes or validates the block. The strategic incentives we highlight will also be present if the role of builder and proposer are separated. Specifically, as the beneficiary of MEV, the builder will have an incentive to interact strategically with any arbitrageurs. We note that, if a builder now has greater flexibility to order transactions, the change is interpretable as an increase in R, or the size of arbitrage opportunities.

3.3 Discussion

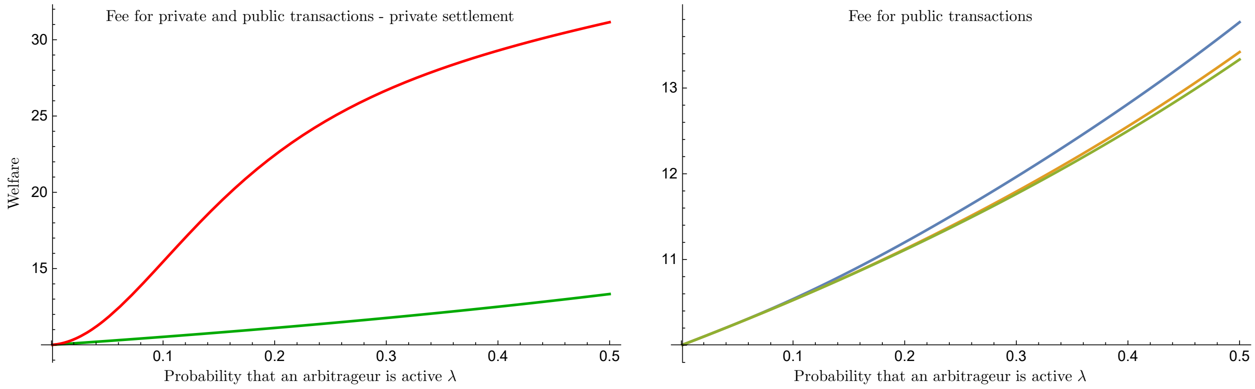

Figure 6. Fees different MEV regimes Fee for private transactions (red) and public transactions (green) when the miner allows for private settlement as a function of λ, the probability that an arbitrageur is active (left panel). The right panel shows fees for public transactions in public settlement when the miner is not a screener (blue), when the miner is a screener (orange), and with private settlement (green). Parameters for the plot are $a = 7, s = 7, R = 40, \bar{v} = 20, \omega = 0.9$

The left panel in Figure 6 compares the fee the miner under private settlement charges for private and public settlement. Under private settlement the miner price discriminates by screening the

[4] see `https://medium.com/flashbots/frontrunning-the-mev-crisis-40629a613752`

[5] A description of some of the properties appears in `https://notes.ethereum.org/@vbuterin/pbs_censorship_resistance`.

mempool and thus charges the arbitrageur a high fee and the liquidity trader a much lower transaction fee. Price discrimination is not perfect, though. A monopolist would charge $\bar{v}/2$ (or 10 in our example) to the liquidity traders. As the arbitrage opportunities become more profitable in expectation (higher λ) the miner first increases the fee for private transactions and increases screening to deter the arbitrageur from moving to the private market. As screening becomes to costly the miner raises the fee in the public market at the expense of loosing some liquidity traders but with the benefit of extracting more from the arbitrageur. The right hand panel shows that under private settlement fees are lowest in the public market. Liquidity traders are therefore better off when a private market exists.

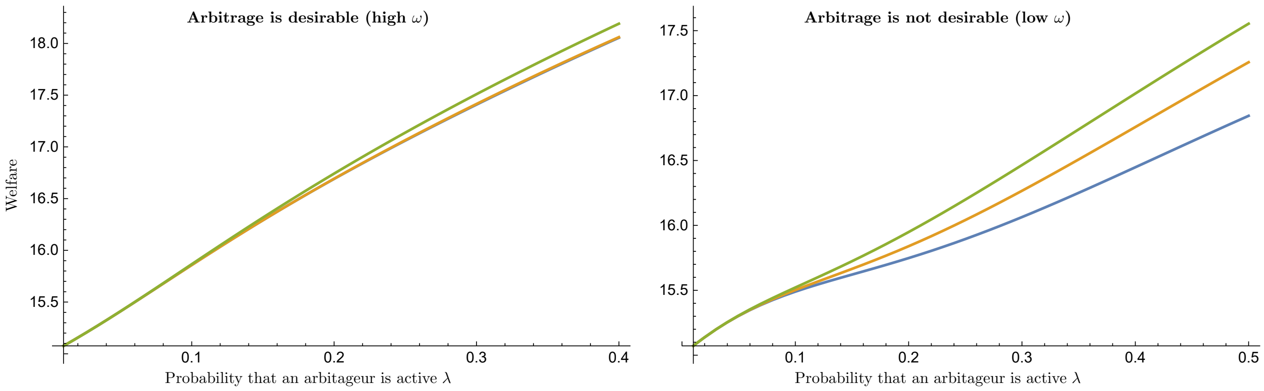

Figure 7. Welfare under different MEV regimes The graph shows welfare as a function of λ, the probability that an arbitrageur is active, for the first best (red), public settlement when the miner is not a screener (blue), when the miner is a screener (orange), and with private settlement (green). The left panel shows a case where arbitrage activity is socially desirable ($\omega = 0.9$) and the right panel where it is undesirable ($\omega = 0.1$). Parameters for the plot are $a = 7, s = 7, R = 40, \bar{v} = 20, \omega = 0.9$

Figure 7 compares welfare under different settlement regimes. When arbitrage activity creates social value (left panel), private settlement improves welfare relative to both public settlement regimes. Welfare is higher under private settlement because no blockspace gets wasted on screening transactions, which provide no social value. A second source of welfare gain in private markets comes from price discrimination. The miner can charge a lower fee for transactions in the public market and thus include more liquidity trades. Yet socially wasteful screening is still performed by the miner to deter the arbitrageur from the public market. In public markets miners who screen internalize the arbitrageur's effort choice better than miners who do not. None of the settlement regimes reach the first best as under all regimes miners' profit maximizing fee will exclude some private value transactions. The social planner sets the fee to zero to include all private value transactions.

The left panel of Figure 8 shows that miner profit is highest when a private market exists. The ability of the miner to price discriminate between arbitrageurs with high value trades and liquidity traders enables the miner to extract more value. The least profitable arrangement for the miner is a market where miners and screeners are separate. The right panel illustrates the profit for the arbitrageur. For a wide range of the parameter space the arbitrageur benefits most

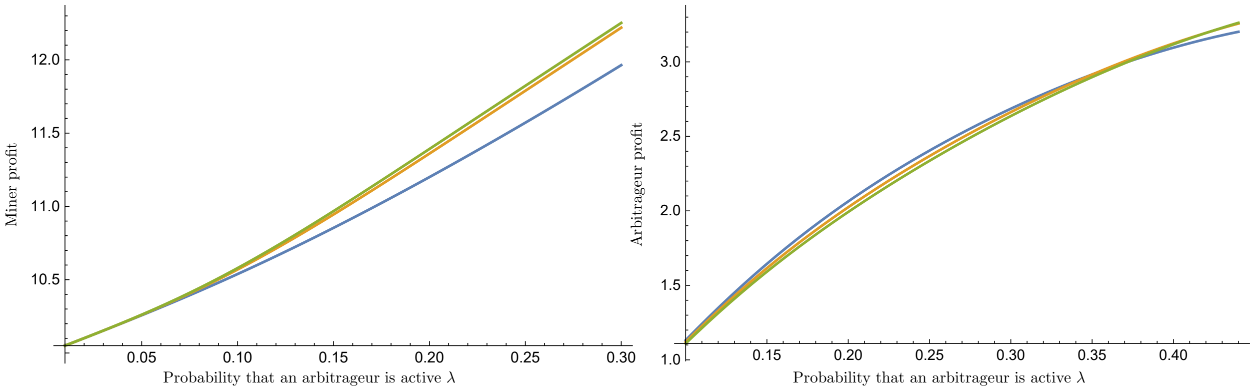

Figure 8. Profit of the miner (left panel) and the arbitrageur (right panel) as a function of λ, the probability that an arbitrageur is active. Plotted for public settlement when the miner is not a screener (blue), when the miner is a screener (orange), and with private settlement (green). Parameters for the plot are $a = 7, s = 7, R = 40, \bar{v} = 20, \omega = 0.9$

in an environment where mining and screening functions are separate.

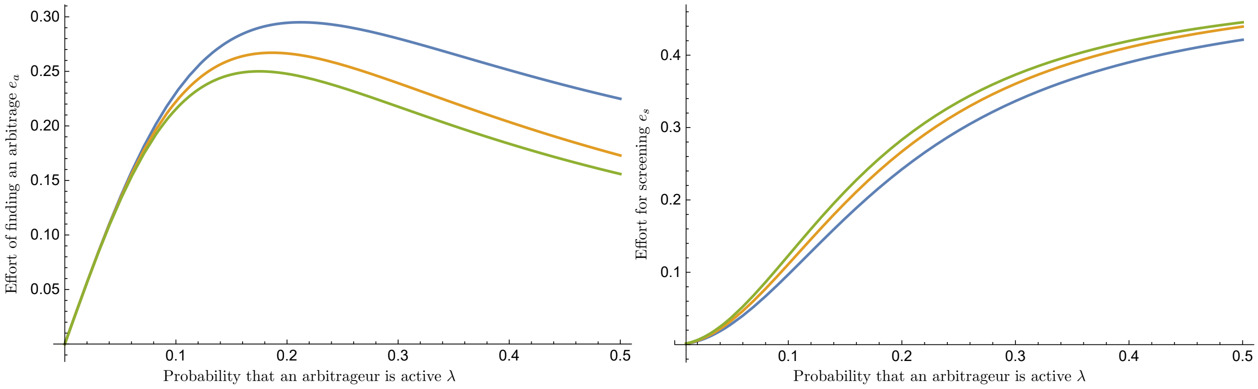

Figure 9. Effort to find an arbitrage opportunity (left panel) and screening effort (right panel) as a function of λ, the probability that an arbitrageur is active. Plotted for public settlement when the miner is not a screener (blue), when the miner is a screener (orange), and with private settlement (green). Parameters for the plot are $a = 7, s = 7, R = 40, \bar{v} = 20, \omega = 0.9$

From the left panel of Figure 9 we can see that private settlement induces the lowest effort level to find arbitrage opportunities. The right hand panel shows optimal screening effort. In the first best there is no screening. Under private settlement miners just have to provide enough screening to make the public mempool unattractive for arbitrageurs. Screening effort is high so that miners can charge spread between public and private markets.

First observe, the fee is lowest if the miner is also a screener and there is no private market.

The welfare for liquidity traders (determined solely by the price they pay for transaction services) is highest if the miner is also a screener and there is no private market.

The private market has the least amount of arbitrage (good or bad) activity, while the highest arbitrage activity occurs in the case in which the miner is not a screener. As we have noted, arbitrage activity can increase or decrease social welfare.

Here, the results are nuanced. If arbitrage activity is not socially desirable (low ω), the private market in which the miner is a screener generates the highest welfare. We note the that the private market when the miner is not a screener is perfectly separating. Conversely, if arbitrage is socially desirable (high ω), the highest welfare coincides with the miner screener with no private market.

The stated purpose of flashbots and in particular the private market was to eliminate MEV on Ethereum. However, if miners are also screeners, then the amount that they can extract from arbitrageurs in the private market depends on the level of screening in the public market. Their incentive is still to screen in the public market, as this is a credible "threat" to extract more surplus from the arbitrageurs in the private market.

4 Stylized facts on Miner Extractable Value

The overall welfare effect of a private market depends on the extent to which there are private trades and the extent to which arbitrage trades are beneficial. In what follows, document arbitrage activity and provide a quantification of beneficial and deleterious arbitrage activity.

Fully quantifying realized and potential MEV is nearly impossible. Miners and users use different ways to their activity that is hard to identify for the econometrician. For example off-chain side payments from users to miners are unobservable. The number of active private market relays is unknown. Potential MEV is even harder to track because of the complexity of potential actions an arbitrageur could initiate. For example an arbitrageur could sell some tokens, pushing the price down, which makes some loans undercollateralized. The arbitrageur could then benefit from liquidating the loans. To find such a trade in billions of potential actions is computationally infeasible. In this section we present some stylized facts that document the importance of MEV and trends over time.

Ethereum transactions allow the execution of code on the Ethereum virtual machine (EVM) that can change the state of the system and thus affect the distribution of wealth between wallets. Therefore the ordering of transactions within a block is not benign. Suppose that an arbitrage opportunity exists between two decentralized exchanges that is spotted by two traders. The trader whose transaction executes first can capture the arbitrage profit while the second trader's transaction will still be mined but will fail because the arbitrage opportunity is gone.

In traditional financial markets transactions can be ordered by timestamps. In a blockchain setting ordering by arrival time is not possible because transactions that wait to be processed are in a decentralized temporary storage, the mempool. Due to network latency and imperfectly synchronized clocks, a precise ordering of transactions based on arrival time is technically not possible. Individual nodes have an incentive to manipulate time stamps.

Miners therefore have complete discretion which transactions to include in a block and how to order them. The fee a user effectively pays is the product of *gas*, a measure of computational complexity, and a gas price, i.e., how many Ether a user is offering per unit of gas. The standard implementation of Ethereum sorts transactions based on the gas price to maximize a miner's revenue. Any blocks where the ordering of transactions is not based on gas price are therefore likely to involve transactions where the miner received some MEV, which could come in the form of either a side-payment or in the form of a transaction that the miner executes on her own behalf.

All Ethereum transactions allow the posting of a gas price which the miner can keep for executing the transaction. The miner can keep this fee regardless if the transaction is successful or not. We find that transactions of bots which submit private transactions such as the one mentioned above usually offer a gas price of zero. Instead of compensating the miner through the regular channel those transactions typically make a direct transfer to the miner via an internal transaction call, which is a wallet to wallet transfer from the bot to the miner.[6]

Our first method to identify private transactions is through unconventional ordering. An example is presented in Figure 10 below. The first transactions incorporated in the block pay a zero gas price, while the remaining transactions are ordered by the gas price. In such transactions remuneration to miners typically occurs via wallet to wallet transfers.

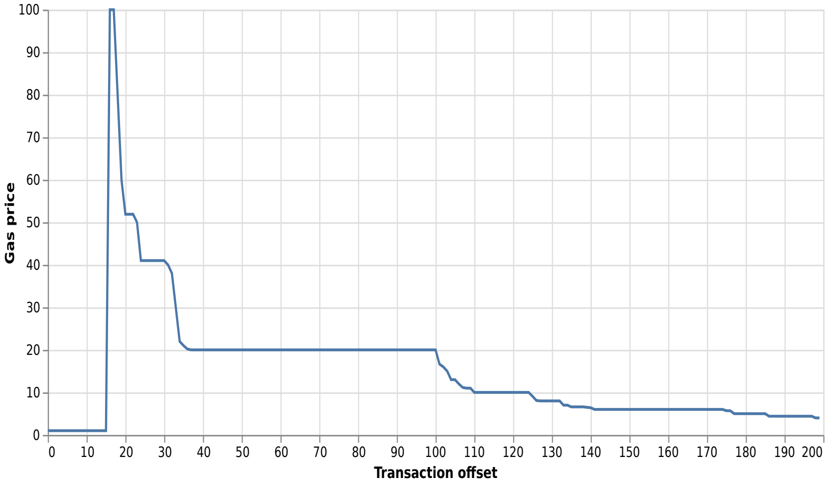

Figure 10. Block 6000003 provides an example of unconventional ordering

We examined all blocks between 6,000,000 and 16,849,737, or July 20, 2018 and March 17, 2023 with a total of 1,632,001,335 transactions for unconventional ordering of transactions. We

[6] One potential reason for this setup is to compensate the miner only in case that the arbitrageur is successful. If the arbitrageur compensated the miner via a gas fee the miner could collect the gas fee whether the transaction is successful or not. If the compensation for the miner gets paid as part of the transaction the arbitrageur pays the miner if and only if the transaction executes successfully, i.e. the miner puts the transaction early in the block to ensure that the trading opportunity still exists.

eliminate 49,587,351 transactions that are initiated by miners. Figure 11 shows the fraction of blocks per day in which transactions are not ordered by gas price. We can see a dramatic increase over time which coincides with the growing concern over MEV but also the rise in the adoption of modified Ethereum nodes that focus on MEV. For our sample we find that on average 59.5% of blocks have an unusual ordering of transactions.

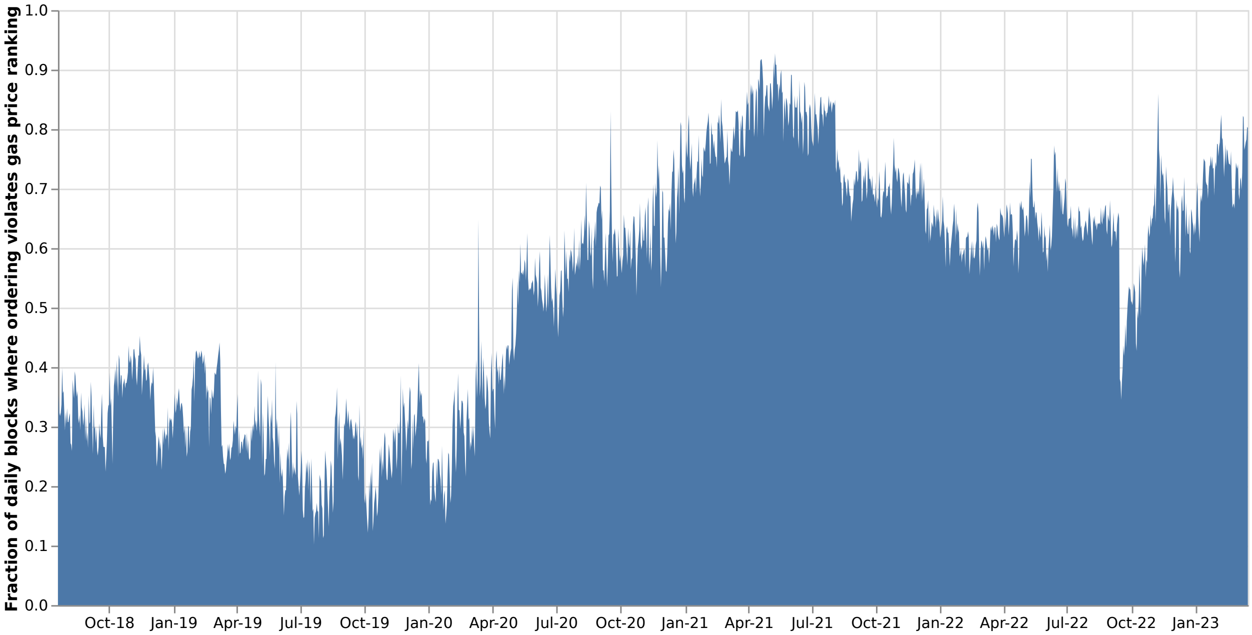

Figure 11. Fraction of blocks with unconventional ordering Fractions of blocks per day for which transactions are not ordered by gas price.

In response to rising concerns on MEV on November 23, 2020 a group called “Flashbots” publicly released MEV-Geth, a fork of the most popular Ethereum implementation GETH with the specific goal “... to propose a permissionless, transparent, and fair ecosystem for MEV extraction that reinforce the Ethereum ideals.” Their software is an “upgrade to the go-ethereum client to enable a sealed-bid block space auction mechanism for communicating transaction order preference”. In other words users who found a profitable trading opportunity, so called seekers, can privately contract with miners to have their transactions included without having to go through the mempool. In return seekers make a payment directly to miners. The package of transactions that seekers propose to a miner can consist of transactions that only originate from the seeker or can also contain transactions that are publicly available in the mempool. Seekers submit ’bundles’ of transactions to Flashbots which can mix private transactions with public transactions that can be found in the mempool. If accepted by the miner the bundle will then be mined as one piece. We will discuss the two examples in detail below.

An example of a transaction that originates from the seeker would be an arbitrage trade between two decentralized exchanges. Suppose that the seeker discovers an opportunity to buy a token at one market and sell it in another market for a profit. If the seeker would post that transaction in the mempool it would become publicly visible and other traders could free ride on the seeker’s effort to find the arbitrage opportunity. Once publicly visible the seeker's transaction could be front run or picked up by a miner who would execute the same trade in their name. To avoid being detected the seeker submits this transactions to a miner directly and proposes a way to share the revenue. If the miner accepts she will put the transaction in front of the block or at least at a position where is will be guaranteed to be executed and collect the fee from the seeker.

A seeker's submission to a miner can also include publicly observable transactions from a mempool. Consider, for example the first three transactions in block 12165347, which are a classic front-running attack.[7] In the middle transaction a trader who's walled is labeled as *Q7 Crypto Fund* trades ETH for RUNE tokens on Uniswap. Like in any market the buy order will increase the price for RUNE tokens. This trader is being front run by a bot who buys 1277.73 Rune tokens for 5.092574 ETH in transaction 1 – just before the Q7's trade. After the price of Rune tokens has increased due to Q7's purchase of 1703.27 RUNE Tokens in transactions 2, the bot sells its rune tokens again fro 5.174446 ETH, making a total profit of 0.081872 ETH or about $164 at the time. That profit is split equally with the miner in a way that is not easily visible on many blockchain explorers[8]

The bot which initiated the front running in the above transaction was active between December 29, 2020 and January 13, 2022, conducted 1,298,402 MEV transactions, and transferred 93,340.7483 ETH or about USD 287.43 million to miners. The bot was working with many different miners which is consistent with the fact that this is an independent seeker and not a miner itself trying to front run individual traders. It also is consistent with many miners using a common interface for seekers to submit private transactions. MEV-Geth provides such a common interface and is assumed to be used by most mining pools today. By submitting a private transaction the bot can ensure that its transaction will be executed first. Indeed 21.86% of the bots transactions are executed as the first three transactions in the block in which they were eventually mined.

We augment our own data with all Flashbots transactions, which are available from their API. Our data comprises all transactions that are submitted through Flashbots and are included in the blockchain. Our sample start on February 11, 2021 with block 11,834,049 and ends on March 17, 2023 and includes 8,048,889 bundles that were submitted by 866,848 wallets. We note that not all users of flashbots are screeners. Some users use private channels to submit their trades or token transfers. For each block we compute miner revenue from flashbot bundles as percentage of overall miner revenue which also includes fees from ordinary transactions. Figure 12 presents daily averages summarizes our findings. On average 15.3% of miner revenue stems from Flashbot MEV transactions.

[7] See transactions `0x3bd5b9f55d120de48330c6e0ac86f68c888724fb86347ad5661f284c71812f27`, `0x620f4fd9e233c2eb13c25db6ffec20ddfe1c3bd2403c97d367d32d935069e332`, and `0xcf9a3e8b59a63c8704a5f2ae656b26fa7420f5ef22906eb21ede036209bb119b`.

[8] In standard Ethereum transactions transfers of the chain's native currency Ether (ETH) are recorded in three entries of the transaction record: 'from' records the sender, 'to' records the receiver, and 'value' records the amount transferred. These fields are provided by a node's standard API and by many data providers. Many transfers that we observe between seekers and miners are recorded as internal transactions and thus not visible in these standardized data interfaces. To collect this information we run an Ethereum archive node and collect a debug trace from replaying old transactions. This debug trace is then filtered for transfers of ETH between the seeker and the miner.

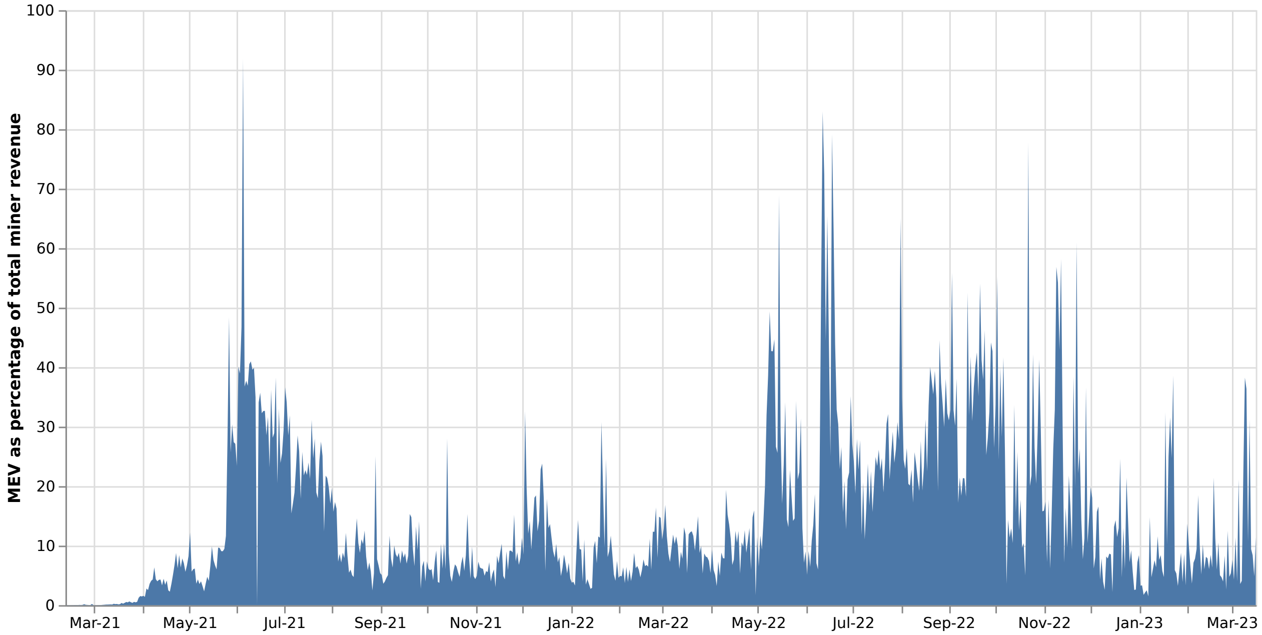

Figure 12. Fraction of daily miner revenue from MEV transactions.

To distinguish good bots from bad bots we look at their typical transaction patterns. Bad activity, such as front running, typically involves three transactions. First the front running bot trades to front run the victim, second the victim trades, and finally the bot unwinds its position. Good bots typically run one or two transactions. Loan liquidations and arbitrage between exchanges require only one transaction. Backrunning transactions consist of a trader that, say trades a token on Uniswap, and moves the price on that Dex while prices on other Dexs remain unchanged. The arbitrageur follows suit and exploits the resulting price difference between Uniswap and other Dexs such as Sushiswap, closing the price gap.

We use these patterns to decompose bot activity. We identify 540,111 good bots which initiate 5,904,835 bundles and 81,673 bad bots which initiate 1,173,504 bundles. We also identify 5,904,835 bundles that do not fit our classification and are labelled as 'other'.

Figure 13 shows the daily transfers from arbitrageurs to miners in USD via the flashbots interface. We see that private transactions create substantial fee revenue for miners. In our sample miners collect on average USD 2,302,641 per day from arbitrageurs. MeV revenue is highly variable and peaks at USD 50,861,631 on August 13, 2022.

Good bot activity is increasing over time and dominates bad bot activity with 70.97% of fees to miners being paid by good bots. This finding is consistent with the rise of platforms that prevent front running such as 1inch and users setting better limits on slippage protection which limits the profitability of front running. Using on chain analysis, detailed in Appendix A, we find that less than 1% of trades on decentralized exchanges are successfully frontrun.

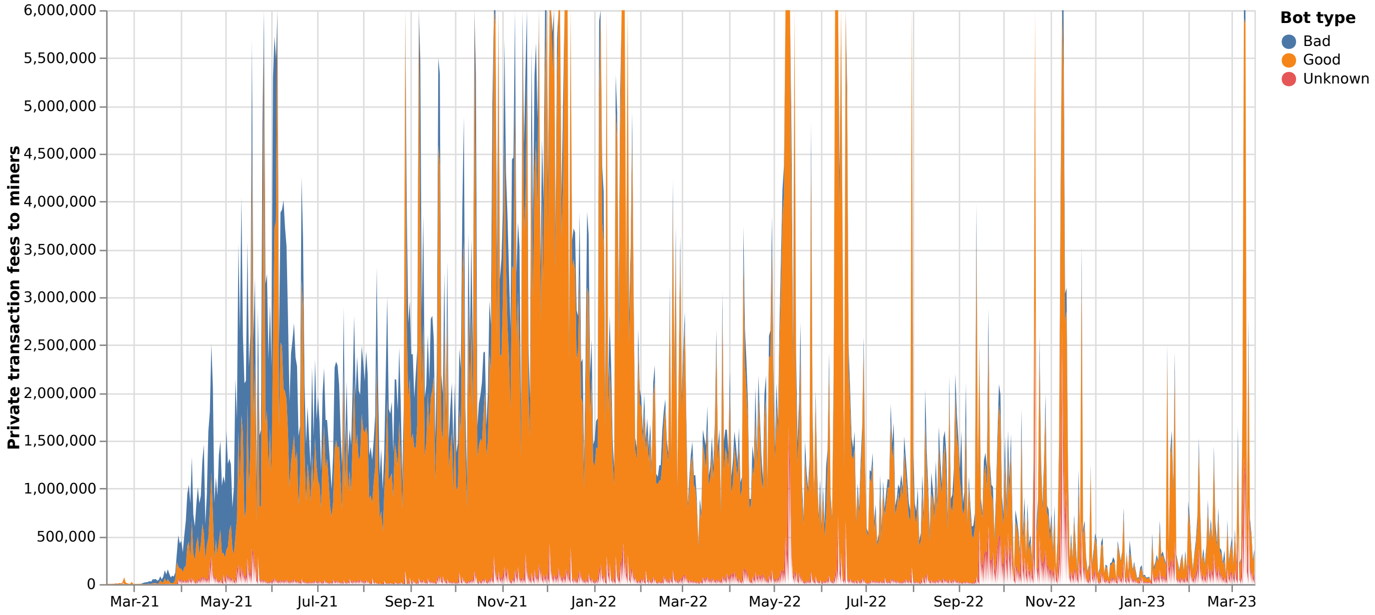

Figure 13. Private transaction fees to miners

In the graph fee payments are capped at USD 6 million.

5 Conclusion

We have presented a framework to understand the tradeoffs inherent in a decentralized, transparent, batch settlement system. On the one hand, the novel system allows arbitrageurs to credibly commit to repay loans from their arbitrage profits. Such flash loans effectively remove any barriers to entry for arbitrageurs. In as much as the DeFi system relies on such traders to ensure that collateral on protocols is sufficient, and to ensure that prices are fresh, this increase in arbitrage activity is good for the DeFi system. On the other hand, the fact that competing settlers can expropriate arbitrageurs' trades inhibits arbitrage activity. Interestingly, the structure of the settlement process in the absence of regulation has implications for both price discovery and stability of the system.

References

Capponi, Augustino, Ruizhe Jia, and Ye Wang, 2021, The Evolution of Blockchain: from Lit to Dark, *Columbia University Working Paper*.

Daian, Philip, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels, 2020, Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability, in *2020 IEEE Symposium on Security and Privacy (SP)* pp. 910–927. IEEE.

Qin, Kaihua, Liyi Zhou, and Arthur Gervais, 2021, Quantifying Blockchain Extractable Value: How dark is the forest?, *arXiv preprint arXiv:2101.05511*.

Zhou, Liyi, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais, 2021, On the just-in-time discovery of profit-generating transactions in defi protocols, *arXiv preprint arXiv:2103.02228*.

A Frontrunning in decentralized exchanges

Because of the visibility of the order flow in the mempool trades on decentralized exchanges are subject to frontrunning risk. In a typical sandwich attack an adversary would, for example, observe a buy order in the mempool and inject her own buy order just in front of the victim's buy, then let the victim's order execute, and then immediately sell her tokens for a profit. These sandwich attacks are fairly easy to implement. A search on Github under 'sandwich bot' reveals 11 public domain repositories, some commercial services are available as well. To test the prevalence of Sandwich attacks we download all Uniswap V2 compatible trades from all exchanges between February 11, 2021 and May 10, 2023. Our sample consists of 95,350,070 swaps in 150,914 liquidity pools.

To filter for basic sandwich attacks we search for patterns of buy-buy-sell or sell-buy-buy orders in the same block, on the same exchange, and where the first order is not more than 50 transactions sway from the last order in the pattern. This classification flags perhaps too many trades as sandwich attack because in reality adversaries tend to put their orders in immediately before and after the victim's order so that the first transaction is typically 2 transactions away from the last one. We also do not require the adversary to use the same wallet address for the frontrunning and the backrunning transaction, even though that is the typical pattern. Overall we find 894,187 patterns that fit a sandwich attack or 0.94% of the sample.

We also match our sample of swaps with all transactions that are bundled in FlashBots. We find that % of all swaps do not appear in a flashbots bundle.

Our findings from this on-chain analysis complement the finding from our main analysis that while important sandwich attacks are not the main component of MEV bot activity.

B Proofs

Proof of Lemma 1

Follows from arguments in the text.

■

Proof of Proposition 1

Part i. and ii. follow from simple algebra.

■

Proof of Proposition 3

Follows from arguments in the text.

■

Proof of Proposition 4(sketch)

Any transfer, $\hat{x} \in [f, f + \widetilde{e_s}R]$ is preferred by the miner and the arbitrageur to the public market. By assumption, arbitrageurs go to the private market when indifferent. For any $\hat{x}$, the arbitrageur will go to the private market. There is no payoff to screening. Thus, $\hat{x} = f$.

■

Proof of Proposition 5

The opportunity cost of the miner is $x = f + \widetilde{e_s}R$. The results follow from the arguments in the text.

■

Proof of Proposition 6

The fee in the presence of a private market is $f = \frac{\bar{v}}{2-\lambda}$. Substituting into the first order condition (6) for the case where the miner is not a screener and only the public market exists yields

$$\frac{(2s(r-\bar{v})^2(r(2-\lambda)-\bar{v})\lambda^4}{as(-2+\lambda)+r(v+r(-2+\lambda))\lambda^2)^2} \tag{16}$$

which is positive as $r - \bar{v} > 0$ by assumption. The miner therefore always charges a higher fee without the private market.

Next examine the case where the miner is a screener and no private market exists. Substituting the fee from the private market case into the first order condition, $\partial\pi_{ms}/\partial f$ from Equation (9) yields

$$\frac{s(r-v)^2(r(2-\lambda)^2-v(1-\lambda))(2-\lambda)\lambda^5}{(as(2-\lambda)^2+r(r(2-\lambda)^2-v(1-\lambda))\lambda^2)^2} \tag{17}$$

which is positive. Again without a private market the miner optimally charges a higher fee.

■

B.1 Flash Loan applications

Several use cases for flash loans are discussed in the computer science literature.

Arbitrage: Several decentralized exchanges like Uniswap, Sushi-Swap, or Balancer are deployed on the Ethereum blockchain that allow trading of token pairs. These exchanges are organized in liquidity pools of two tokens that allow the exchange of one token against another one using an automated market making (AMM) mechanism. These AMMs often quote stale prices as they cannot observe quoted prices at centralized exchanges that are outside of the blockchain and it is often to expensive (and risky) to obtain quoted prices from other AMMs.[9] Instead decentralized exchanges rely on arbitrageurs to bring prices back to equilibrium. Arbitrage opportunities can arise between two different exchanges that trade the same token pairs or as triangular arbitrage involving three different liquidity pools (e.g., converting token A to B, B to C, and then C to A for a profit). Traders can use flash loans to take advantage of arbitrage opportunities without having to invest their own capital. Towards the end of our sample period about 17 billion USD is invested in liquidity pools of decentralized exchanges.

Collateral change: The largest share of capital in DeFi , about 19 billion USD at the end of our sample, is allocated to lending platforms such as Maker, Compound, or Aave. In these pools investors can contribute towards a lending pool from which borrowers can draw collateralized loans. Both the loan and the collateral are typically tokens. A popular trade is to build a levered position in ETH by buying ETH, posting it as collateral on such a platform in return for USD stablecoins and then swapping the USD stablecoins for more ETH. Users who want to swap their collateral face a funding need because due to way smart contracts are implemented in Ethereum new collateral has to be deposited before old collateral can be released to the borrower. Borrowers can borrow the new collateral using a flash loan, release the old collateral, and use a decentralized exchange to convert the old collateral to the denomination of the loan, and repay the loan in one transaction.

Loan liquidation: When the collateral of the loans falls below the liquidation threshold the loans can be liquidated. A liquidator, often a bot, can repay the loan and seize the collateral, often at a discount relative to current market values. In a typical liquidation transactions a liquidator takes out a flash loan to pay the lending platform, seizes the collateral, converts the collateral to the denomination of the flash loan on a decentralized exchange, and repays the flash loan with the proceeds. As mentioned above the liquidator has an option like payoff because the transaction will only execute if the proceeds from the sale of the collateral exceed the amount of the flash loan.

Exploits: The most spectacular and widely reported use cases are the exploitation of weaknesses in other DeFi protocols. Such exploits are often referred to as hacks although no hacking is involved. Exploits are possible because of poorly programmed smart contracts. An early and well publicized attack occurred on February 15, 2020 when the lending protocol bZx lost approximately USD 620,000 in a complex attack.[10] An attacker borrowed 10,000 ETH in a flash loan from dYdX and used about half to open a 5x levered position on bZx shorting ETH vs BTC. To hedge the position bZx automatically placed a huge order on Uniswap selling ETH for BTC, thus driving down the ETH/BTC exchange rate. With the second half of the flash loan the attacker took advantage of the depressed price

[9] One particular decentralized exchange has no way of knowing whether its price is correct or that of another exchange is correct. If an exchange mimics quotes on another exchange, hackers could try to manipulate prices at other exchange strategically to trade at prices that work in their favor.

[10] see transaction 0xb5c8bd9430b6cc87a0e2fe110ece6bf527fa4f170a4bc8cd032f768fc5219838.

and bought ETH form Uniswap at below market prices. Due to a mistake in the code of bZx the position was undercollateralized and the attacker could walk away from his levered position with a profit of approximately USD 370,000. On November 14, 2020 an attacker exploited a weakness in the code of ‘value DeFi’ causing a loss of 8 million USD. The platform boasted on November 13 that it had the highest security and was immune to flash loan attacks. A day later, using two flashloans from Aave and Uniswap for a total of 150 million USD, an attacker exploited 8 million USD from value DeFi and returned 2 million with a message “do you really know flash loan?”.[11]

[11] See transaction 0x46a03488247425f845e444b9c10b52ba3c14927c687d38287c0faddc7471150a for the attack and the input data of transaction 0x217298bd38ed12b16e0cd65ce0b464c3810e0479a99a1464aed5e6768b2a4c50 for the message.