# A Flash(bot) in the Pan: Measuring Maximal Extractable Value in Private Pools

Ben Weintraub*
Northeastern University
weintraub.b@northeastern.edu

Christof Ferreira Torres*
SnT, University of Luxembourg
christof.torres@uni.lu

Cristina Nita-Rotaru
Northeastern University
c.nitarotaru@northeastern.edu

Radu State
SnT, University of Luxembourg
radu.state@uni.lu

## ABSTRACT

The rise of Ethereum has lead to a flourishing decentralized marketplace that has, unfortunately, fallen victim to frontrunning and Maximal Extractable Value (MEV) activities, where savvy participants game transaction orderings *within a block* for profit. One popular solution to address such behavior is Flashbots, a private pool with infrastructure and design goals aimed at eliminating the negative externalities associated with MEV. While Flashbots has established laudable goals to address MEV behavior, no evidence has been provided to show that these goals are achieved in practice.

In this paper, we measure the popularity of Flashbots and evaluate if it is meeting its chartered goals. We find that (1) Flashbots miners account for over 99.9 % of the hashing power in the Ethereum network, (2) powerful miners are making more than 2× what they were making prior to using Flashbots, while non-miners' slice of the pie has shrunk commensurately, (3) mining is just as centralized as it was prior to Flashbots with more than 90 % of Flashbots blocks coming from just two miners, and (4) while more than 80 % of MEV extraction in Ethereum is happening through Flashbots, 13.2 % is coming from other private pools.

## CCS CONCEPTS

• **Security and privacy** → *Domain-specific security and privacy architectures*; Security protocols.

**ACM Reference Format:**
Ben Weintraub, Christof Ferreira Torres, Cristina Nita-Rotaru, and Radu State. 2022. A Flash(bot) in the Pan: Measuring Maximal Extractable Value in Private Pools. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3517745.3561448

## 1 INTRODUCTION

Many investors find the promise of Decentralized Finance (DeFi) appealing. This has lead to cryptocurrency exchanges seeing over 78 billion USD in daily trade volume [20]. Meanwhile, cryptocurrency donations to Ukraine surpassed 63 million USD between February 26th and March 11th, 2022 [24], indicating the usefulness of cryptocurrencies for both private investors and nation states.

Unfortunately, the exchanges in which these currencies are traded, and especially those running on the Ethereum blockchain [55], are victim to a costly type of malicious behavior: *frontrunning*. Frontrunning is a concept borrowed from traditional finance and popularized by Michael Lewis's bestselling *Flashboys: A Wall Street Revolt* [45]. Frontrunning is a type of high-frequency trading where one party uses privileged access to pools of pending trades to execute their own trade before a targeted pending trade. There are several types of frontrunning, but all rely on extracting profit from slippage caused by a publicly viewable transaction.

Daian et al. [33] found, in 2020, that frontrunning analogous to traditional markets is plaguing cryptocurrency exchanges: particularly decentralized exchanges on Ethereum. Frontrunning in the cryptocurrency world is also known as *Maximal Extractable Value* (MEV); it occurs when a malicious peer on the blockchain network learns of an uncommitted transaction and is able to exert some control of the transaction ordering for its own profit [33]. MEV is a widespread problem. Torres et al. [53] found that 199,724 frontruns occurred from July 30th, 2015 to November 21th, 2020, for a cumulative profit of 18.4 million USD.

Since the pioneering work of Daian et al. [33], the research community has published numerous solutions for addressing this problem [12, 39, 40, 43]. However, the most popular solution, in practice, is called *Flashbots* [13]. Flashbots works by creating a private transaction pool where non-mining participants (called *searchers* in Flashbots parlance) submit immutable bundles of transactions to relays who broadcast them to participating miners. The miners then mine the most profitable bundles with each block. Searchers must pay miners a fee in exchange for including their transactions in blocks. Flashbots is not strictly a defense against MEV. It allows transaction submitters to control transaction ordering, which means that *anyone* can now be an MEV extractor. The only defense it offers is that submitters who do not want to be frontrun can also submit their transactions through Flashbots. This offers them protection, because the transaction ordering is guaranteed by Flashbots.

Despite a number of structural flaws, Flashbots adoption skyrocketed over the course of 2021. To date, however, this solution has not been audited by external researchers. In addition to Flashbots, there are other private transaction pools. The relationship of

*Both authors contributed equally to the paper.


*IMC '22, October 25–27, 2022, Nice, France*

ACM ISBN 978-1-4503-9259-4/22/10…$15.00
https://doi.org/10.1145/3517745.3561448

these pools (and users thereof) to Flashbots is, to our knowledge, unexplored in the literature.

In this paper we measure and analyze the impact of Flashbots, and evaluate whether or not it is meeting its claimed goals. We also evaluate non-Flashbots private pools with a focus on MEV extraction occurring in those pools. In particular, we highlight three key findings.

★ Flashbots usage grew rapidly at first, and by now has captured nearly all of the Ethereum network's hashing power. Only five months after its release, Flashbots had already reached a hash rate of 97.6 % and was hovering around 99.9 % after only eight months.

★ Flashbots is disproportionately beneficial for miners at the expense of non-miners. In addition, miners using Flashbots are making more profit than before using Flashbots, while non-miners are making less. This has impacted usage, which has driven an exodus from the system by some users.

★ Not all MEV extractors are using Flashbots. There is a small but powerful contingent using other private pools. In some cases, the pools consist of a single miner.

The rest of the paper proceeds as follows. We first give necessary background on DeFi and the affected exchanges (Section 2). Next, we describe the data we collected and analyzed (Section 3). We then move onto a discussion on the usage of Flashbots (Section 4) and an audit on how well it is achieving its goals (Section 5). In Section 6, we describe and analyze other private pools in Ethereum, and how they are involved in MEV, followed by related work in Section 7. We conclude in Section 8 with a discussion on the efficacy of Flashbots as an MEV solution, and whether or not it is a viable long-term solution.

# 2 BACKGROUND

## 2.1 Ethereum

Ethereum [55] is a decentralized blockchain that facilitates cryptocurrency transactions and *smart contract* execution. The blockchain consists of blocks, which themselves consist of *transactions*. Ethereum nodes propagate transactions through the network where *miners* collate them into blocks. Transactions received over the network are collected locally at each participating node in the *mempool*. The mempool has no blockchain-like guarantees of consistency.

Miners select a subset of the mempool transactions and execute a *proof-of-work* challenge [48]. Whichever miner completes the challenge propagates the block to the rest of the network where each node adds it to their local view of the blockchain. Nodes that submit transactions but do not mine blocks, are called *non-miners*. Note that while the purpose of miners is to mine blocks, they may also submit and mine their own transactions.

The ordering of transactions within a block is important. Each transaction is crafted to operate on a specific blockchain state, but when a transaction executes, it changes this state. Therefore, a transaction's execution can depend on the set of transactions preceding it both in previous blocks and within the same block. Miners have full discretion as to intra-block ordering. The default strategy is to sort pending transactions in the mempool in descending order by fees-per-byte. This strategy optimizes profit when transactions are considered individually.

A transaction may include a *smart contract* [52]; an executable set of instructions for the Ethereum Virtual Machine (EVM). A smart contract will often specify the terms of a cryptocurrency transaction (e.g. source, destination, amount, conditions, etc.), but may also specify arbitrarily complex programs.

As the EVM instruction set is Turing-complete [55], Ethereum introduces the notion of *gas* to guarantee program termination. Gas works by charging a fee per executed instruction. The cost of an instruction in *gas* is constant, however, when a transaction is submitted to the network, the submitter must choose an exchange rate of Ether (ETH) to gas. This is called the *gas price*. Transaction submitters are incentivized to choose competitive gas prices so that miners select their transactions when mining blocks. Miners run the smart contracts, and, upon successfully mining a block, receive, as a fee, the gas used for each transaction within the block. If a contract runs out of gas, the miner gets to keep the gas fees, but rolls back any side-effects of the contract.

## 2.2 DEXes, Frontrunning, and MEV

Like a traditional market, cryptocurrency markets rely on exchanges. An exchange is somewhere traders can swap assets, and usually offers some form of protection for the involved traders. A popular type of exchange for cryptocurrencies is a *decentralized exchange* (DEX). In a DEX, the logic of the exchange is relegated to a smart contract. In one type of DEX, called an *Automated Market Maker* (AMM), the DEX contracts hold currency reserves themselves instead of being maintained by a trusted bookkeeper.

For decades, investors in traditional finance markets have sought creative modes of profit that give them a competitive advantage. One such method is called *frontrunning* [29]. Frontrunning is a predatory investment strategy where an investor learns of an attempted trade in an exchange by a third party. Then uses privileged access to the trading platform to execute trades faster than their victim.

*2.2.1 Transaction Ordering.* In the context of Ethereum, a miner can engage in frontrunning by simply choosing to place one of their own transactions ahead of the target victim within the same block. A non-miner can, in fact, achieve the same effect by increasing the gas price such that a rational miner will naturally order the non-miner's transaction ahead of the victim. The inverse of this is *backrunning* in which the MEV extractor wishes their transaction to be ordered after the victim. Backrunning can be accomplished by analogous methods for both miners and non-miners.

*2.2.2 MEV Extraction.* In their seminal 2020 paper, Daian et al. [33] showed that frontrunning and backrunning are common occurrences in Ethereum DEXes along with several other types of predatory transaction behaviors. They collectively called these behaviors *Miner Extractable Value* (later changed to *Maximal* Extractable Value, and usually referred to as MEV). The idea behind MEV extraction is for profit seekers to discover financial instabilities in existing protocols and to craft transactions that exploit these instabilities in order to extract monetary value. MEV extraction may leverage and combine transaction ordering primitives.

MEV in the DeFi space is made possible by the incentives behind transaction ordering within blocks. In this paper, we will discuss

three MEV strategies, and their impacts. The strategies are *sandwiching*, *arbitrage MEV*, and *liquidation MEV* [51, 58].

**Sandwiching.** Sandwiching is a classic trading strategy and is well-known in the traditional financial world. However, it can be also be leveraged for MEV extraction. An MEV extractor starts by monitoring the mempool for pending transactions that are about to trade large sums of a particular asset. A large transaction will result in a fluctuation in the price of the asset. Knowing this, the MEV extractor then crafts a so-called "sandwich", by surrounding this large transaction with two of its own transactions. In the first transaction, the MEV extractor frontruns the large transaction in order to buy or sell some quantity of the asset before the price of the asset fluctuates. In the second transaction, the MEV extractor backruns the large transaction in order to either buy back the original asset at a lower price, or sell the newly acquired asset for a higher price. In both cases the MEV extractor makes a profit due to the price difference.

We model sandwiching as follows.

**Definition 1.** Consider a victim transaction $V$ and sandwich transactions $t_1$ and $t_2$ originating from the same address. Transactions $t_1$ and $V$ are exchanges from currency $C_x$ to currency $C_y$, and $t_2$ is an exchange from $C_y$ to $C_x$. All three transactions are within the same block $B$, i.e., $t_1, t_2, V \in B$. A sandwich MEV is said to have occurred if $t_1 \prec V \prec t_2$.

**Arbitrage MEV.** Arbitrage is the process of trading assets simultaneously across different exchanges in order to profit from price differences; it is typically considered benign as it balances out price differences across exchanges and keeps the market stable.

An MEV extractor can perform MEV on arbitrage transactions by either following a *passive* or *proactive* strategy. When following a passive strategy, the MEV extractor monitors the current blockchain state and compares prices for different assets across multiple exchanges and only executes an arbitrage if the expected revenue of buying an asset on one exchange and selling it on another exceeds the expected transaction costs.

In the proactive strategy, the MEV extractor monitors the mempool seeking either a pending arbitrage transaction or a large pending trade. When a pending arbitrage transaction is detected, the MEV extractor simply copies the transaction and pays higher transaction fees in order to frontrun the existing transaction and claim the profits of the arbitrage transaction. When a large pending trade is spotted, the MEV extractor will first check if the large trade will result in a price difference across different exchanges and only then craft an arbitrage transaction around the large trade by attempting to backrun the large pending trade with its own arbitrage transaction.

We formally define proactive arbitrage MEV as follows.

**Definition 2.** Consider a victim arbitrage transaction $V$ that is unpublished and has only been propagated in the public mempool. $V$ is an exchange of a single currency $C$ between two exchanges $E_1$ and $E_2$, we write this as $E_1(C) \rightarrow E_2(C)$. We denote the price of $C$ on an exchange as $P(E_n, C)$. Arbitrage can occur if $P(E_1, C) + F_{vic} < P(E_2, C)$, where $F_{vic}$ is the mining fee paid by the victim. If an MEV extractor learns of such an opportunity in the mempool, it can submit the same transaction $E_1(C) \rightarrow E_2(C)$ with a higher fee (i.e., $F_{mev} > F_{vic}$) as long as $P(E_1, C) + F_{mev} < P(E_2, C)$ still holds.

**Liquidation MEV.** Lending and borrowing assets is one popular use case of DeFi. Lenders aim to profit by charging interest on capital they lend to *lending pools* (e.g. Aave [7] or Compound [4]). Borrowers borrow assets from these lending pools by agreeing to pay interest on the loan and by proffering a security in the form of collateral worth more than the borrowed asset.

Complicating matters, the value of a collateral can fluctuate over time. If the price of the collateral drops by a certain critical value, it becomes worth less than the value of the borrowed asset. In order to prevent the collateral from reaching this critical value, lending pools allow anyone to *liquidate* loans. Liquidation is when a user (distinct from the borrower) repays the debt on behalf of the borrower in exchange for receiving the collateral at a discounted price. The discount on the collateral incentivizes liquidation, so the lender can avoid losing money on the loan. When the value of a collateral approaches the critical value, the lending pool marks the loan as "unhealthy" and makes the loan available for liquidation. After liquidation, the loan becomes "healthy" again, and the liquidator can profit by selling the collateral in the market.

There exist two different types of liquidation mechanisms: *fixed spread-based* liquidations and *auction-based* liquidations. Fixed spread-based liquidations are settled within one blockchain transaction and follow the first-come-first-served principle: whoever offers to liquidate a loan first receives the collateral. On the other hand, auction-based liquidations are initiated by interested liquidators who provide bids with the highest bid receiving the loan's collateral. An auction may last several hours and is non-atomic as it may require liquidators to interact with the lending platform via multiple transactions.

Due to their atomicity, fixed spread-based liquidations are a prime target for MEV extraction. Similar to arbitrage, a would-be MEV extractor can either follow a *passive* or *proactive* strategy to perform liquidations. When following a passive strategy, the MEV extractor only scans the current blockchain state for liquidation opportunities and attempts to frontrun competing liquidators. In the proactive strategy, the MEV extractor monitors the mempool for either of two types of opportunities. The first are pending liquidation transactions, which the extractor can then copy and frontrun; the second are pending transactions that will create a liquidation opportunity when mined (e.g., an oracle price update that renders collateral open for liquidation), for which the extractor then crafts a liquidation transaction and backruns the transaction that created the liquidation opportunity.

We formally describe fixed spread-based, proactive liquidation as follows.

**Definition 3.** Consider a collateral $C$ which is leveraged against a loan with value $v_L$. The current price of the collateral at time $t$ is $P(C, t)$. When the loan is taken out, $P(C, t_0) = v_L$. If at some future time $P(C, t_n) < v_L$, the loan is released by the lending pool for liquidation.

An MEV extractor may attempt to frontrun a liquidation $tx_1$ by submitting a transaction $tx_1'$ (a copy of $tx_1$) where $tx_1' \prec tx_1$ in the next block. As in sandwiching, this is accomplished by setting $tx_1'.fee > tx_1.fee$.

## 2.3 Flash Loans

Flash loans are a concept unique to DeFi. They are loans taken and repaid within the same transaction. Flash loans are possible because a single transaction can provide guarantees on the collective execution of multiple smart contract functions, i.e., they can ensure that either all function calls succeed or the transaction is reverted. The advantage of Flash loans is that they allow users to borrow any assets available in a lending pool without requiring any collateral. If the user is not able to pay back the loan plus interest at the end of the execution of the transaction, then the transaction is reverted, meaning any state changes that occurred during execution are rolled back.

As flash loans are bound to the execution of a single transaction, they cannot be leveraged by MEV extractors for sandwiching. However, they can be leveraged by MEV extractors to perform arbitrage MEV and liquidation MEV using large, borrowed assets. The MEV extractors are only required to own enough assets to pay back the interest and execution costs.

## 2.4 Private Transactions

In Ethereum, most transactions are propagated when users submit their transactions to Ethereum nodes which then gossip them to other nodes in the network [41]. Eventually, every node that is part of Ethereum's peer-to-peer network will receive the transaction.

In Ethereum, transactions are not encrypted, so everyone can inspect the contents of a transaction. This, combined with the fact that transactions are publicly propagated, allows users to simply connect to an Ethereum node and monitor which transactions are pending and analyze their purpose. This openness, combined with the fact that miners predictably order their transactions by gas price, ultimately enables frontrunning and MEV extraction.

One way to mitigate these complexities is to send transactions directly to a trusted miners who will not further propagate the transaction and will mine it secretly. These secretly submitted transactions are called *private transactions*. Projects such as Flashbots or the Eden Network [49] try to optimize this process of creating private agreements with miners and establishing so-called *private pools*. Transactions within these pools are only visible and forwarded to trusted miners.

There are typically three reasons why users might want to use private transactions: miner payouts, transaction privacy, and frontrunning/backrunning. Private transactions are useful for miner payouts in mining pools because their pool does not need to include any transaction fees in its payout transaction. It can get away with this because it knows that all miners in the mining pool have a vested interest in the transaction being included. Private transactions are also useful for maintaining some degree of transaction privacy. Users may want to hide the intention of their transactions, for example, when performing a trade they do not want to be affected by MEV extraction, or when performing MEV extraction they do not want to be frontrun, themselves, by competitors. Finally, private transactions are useful for would-be frontrunners who wish to maintain their competive advantage by hiding their practices from other frontrunners.

## 2.5 Flashbots

Flashbots is the name of an ecosystem of projects with the stated goal of limiting frontrunning's negative externalities. It does this primarily through two initiatives: `MEV-geth` [14], and `MEV-inspect` [15].

The Flashbots ecosystem works by creating a private transaction pool, which can only be accessed by nodes assuming one of three delineated roles. The roles are: *searchers*, *relays*, and *miners* (Figure 1). Searchers are non-miners who listen for public transactions propagated by peers (Step ❶). When they identify MEV opportunities, they build a *bundle* of transactions, which is an immutable, atomic set of transactions—either all transactions in the bundle are executed *in order*, or none of them are. Bundles are a concept exclusive to Flashbots. The bundle includes a fee that is paid out to the miner, but the MEV profit from the submitting searcher's transactions are kept by that searcher. There are three different bundle types: (1) *miner payout*, which include transactions used by the miner to pay users of a mining pool, (2) *rogue*, which include transactions introduced by the miner and not broadcast even with Flashbots, and (3) *flashbots*, which follow the standard dataflow described below.

After the searcher has identified MEV opportunities, they then forward the bundle to the relays (Step ❷) instead of gossiping to the public mempool as is the case with non-Flashbots transactions. Flashbots bundles (and their constituent transactions) thus remain visible only to Flashbots miners; they become visible to the rest of the public Ethereum network only after they have been mined into a block. Relays collect bundles from searchers and forward them to miners—they exist primarily as DoS protection for unprepared miners. Flashbots plans to add the ability for any node to serve as a relay in the future, but currently, there is only one relay in the system, run by the Flashbots project itself.

Miners collect bundles from relays (Step ❸), and mine whichever bundles are most profitable for them—only a single bundle can be included per block to guarantee the searcher's intent is satisfied. Since bundles are just an ordered set of transactions, they do not qualitatively differ from any other Ethereum transactions after mining. Miners can participate by running the Flashbots-provided `MEV-geth`—a fork of the reference go-ethereum (`geth`) client [16]. Miners can choose which bundles to mine, most likely based on potential for profit, and they can choose not to mine bundles at all. All choices for miners are devoid of pecuniary risk as MEV is a pure profit opportunity for them. However, if they choose to mine a bundle, they cannot in any way modify that bundle. Equivocating on a bundle leads to a permanent ban from using the Flashbots infrastructure.

Flashbots is able to threaten banning, because they control access to the system. Any miner or searcher wishing to participate in Flashbots must first apply through the Flashbots web portal. They supposedly permit anyone to join who agrees to honor the above invariants, however this has not been empirically evaluated. Upon passing a review by the Flashbots project, a miner is authorized to receive bundles from the single operational relay. Abuse of the system rules will result in revocation of a miner's authorization.

*2.5.1 Goals of Flashbots.* Flashbots was specifically designed with three goals in mind. All design decisions are allegedly in service of these three goals. We describe them here in no particular order.

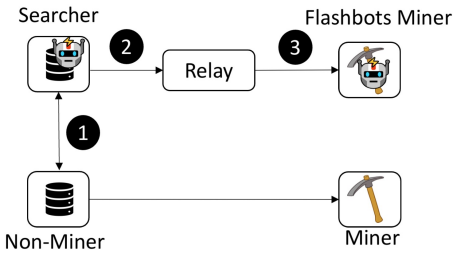



**Figure 1: Flashbots architecture. Searchers, a subset of non-miners, send transaction bundles to relays who then forward them to participating Flashbots miners. Non-Flashbots nodes are unaware of these bundles.**

**Goal 1** (Illuminate the Dark Forest)**.** Flashbots claims that the MEV problem is difficult to measure and quantify. As such, they are spearheading a number of measurement initiatives [17, 18]. The end goal: more transparency into how much MEV is occurring, and to what degree.

**Goal 2** (Democratize MEV Extraction)**.** Form a neutral, public, open-source, and permissionless infrastructure for MEV extraction, with the intent that MEV extraction can be done by any node in Ethereum—not just those that can afford expensive infrastructure.

**Goal 3** (Distribute Benefits)**.** MEV primarily benefits traders and miners, but involves most participants in the Ethereum ecosystem. This goal of Flashbots is to help all participants in Ethereum profit from MEV.

## 3 DATA COLLECTION AND PROCESSING

To evaluate the efficacy of Flashbots as a solution to the MEV crisis, we adopt a data-driven methodology. We describe our data collection in this section as well as our methods for extracting insights from these data. In the service of open science, we make our datasets and collection code openly available[1].

Figure 2 provides an overview of our measurement setup for collecting MEV-related data as well as pending transactions on the network. Some of our data was collected using an Ethereum *archive node*, which we setup using `go-ethereum` [16]. An archive node provides a complete history of all state changes on the Ethereum blockchain, which allowed us to query data on any published block.

Our data collection focused on the block range between 10,000,000 and 14,444,725 (May 4th, 2020 to March 23rd, 2022). All data collection used a machine with 18 TB SSD storage, 128 GB memory and 10 Intel (R) Xeon (TM) L5640 CPUs with 12 cores each and clocked at 2.26 GHz, running 64-bit Ubuntu 16.04.6 LTS. We store our collected data inside a MongoDB database.

### 3.1 MEV

We developed scripts that leverage heuristics presented by previous works [51, 53] to measure extracted MEV. We also leveraged techniques described by Wang et al. [54] to detect flash loans and use public Flashbots data to identify if the MEV has been extracted via Flashbots.

[1]https://github.com/a-flashbot-in-the-pan

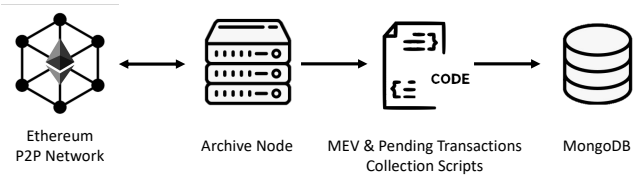



**Figure 2: Measurement architecture.**

| MEV Strategy | Extractions | Via Flashbots | Via Flash Loans | Via Both |
|---|---|---|---|---|
| Sandwiching | 1,020,044 | 485,680 (47.61%) | 0 (0%) | 0 (0%) |
| Arbitrage | 3,462,678 | 916,709 (26.47%) | 10.155 (0.29%) | 1.089 (0.03%) |
| Liquidation | 32,819 | 9,191 (28.01%) | 1.672 (5.09%) | 132 (0.4%) |
| **Total** | **4.515.541** | **1.411.580 (31.26%)** | **11.827 (0.26%)** | **1.221 (0.03%)** |

**Table 1: MEV dataset overview.**

*3.1.1 Sandwiches.* We measured sandwiches by crawling token transfer events from our archive node and applying the heuristics developed by Torres et al. [53] for detecting insertion frontrunning (i.e., sandwiching). The heuristics by Torres et al. [53] assume that the attacker buys and sells the same type of tokens as the victim across two different transactions where the amount purchased and sold by the attacker is almost identical and where the gas price of the attacker's first transaction is higher than the victim's transaction. Our script is capable of detecting sandwiches across the following exchanges: Bancor [2], SushiSwap [8], and Uniswap V1, V2 and V3 [3].

We computed the profit that the MEV extractor made by deducting the costs of the sandwich transactions from the gain made through those sandwich transactions. The costs were computed as the sum of the transaction fees for both sandwich transactions and any tips that the MEV extractor payed to the miner via coinbase transfers (these tips only apply when the MEV extractor used Flashbots).

The gain was computed as the difference between the ether used to purchase the tokens and the ether obtained from selling the tokens. We collected 1,020,044 sandwich attacks, where 485,680 (47.61 %) of these sandwich attacks were performed by using Flashbots (see Table 1).

*3.1.2 Arbitrage.* We measured arbitrage MEVs by crawling token swap events from our archive node and applying the heuristics developed by Qin et al. [51]. The heuristics by Qin et al. [51] assume that an arbitrage must include more than one swap event and that all swap events must be included in a single transaction thereby forming a closed loop. Our script detected arbitrages across the following exchanges: 0x Protocol [1], Balancer [5], Bancor [2], Curve [9], SushiSwap [8], and Uniswap V2 and V3 [3]. We compute the profit that the MEV extractor made, by deducting the costs of the arbitrage transaction from the gain made through the arbitrage transaction.

The costs were computed as the sum of the transaction fees for the arbitrage MEV transaction and any tips that the MEV extractor paid to the miner via coinbase transfers (these tips only apply when the MEV extractor used Flashbots).

The gain was computed as the sum of the assets that were left after the swaps were executed. Assets may include ether or tokens. To keep our analysis consistent, we convert the value of the tokens to ether. To do so, we leverage CoinGecko's API [19]. We collected 3,462,678 arbitrages, where 916,709 (26.47 %) of these arbitrages were performed by using Flashbots, 10,155 (0.29 %) by using flash loans, and 1,089 (0.03 %) by using both Flashbots and flash loans (see Table 1).

*3.1.3 Liquidation.* We measured liquidation MEVs by crawling liquidation events from our archive node for different lending platforms and extracting information from these events such as liquidated debt and received collateral. Our script is capable of detecting liquidations across the following lending platforms: Aave V1 and V2 [7], and Compound [4]. We did not rely on previous works to detect liquidations. Concretely, our script searches for Aave's `LiquidationCall` event and Compounds's `LiquidateBorrow` event, which directly correspond to liquidations.

We computed the MEV extractor's profit by deducting the costs of the liquidation MEV transaction from the gain made through the liquidation transaction. Like in the arbitrage case, the costs were computed as the sum of the transaction fees of the liquidation MEV transaction, the value of the liquidated debt, and any tips that the MEV extractor payed to the miner via coinbase transfers (the last one only applies if the MEV extractor performed the liquidation through Flashbots.

The gain was computed as the value of the received collateral. Similar to arbitrage, we converted all collateral into its equivalent amount in ether using the CoinGecko's API. We collected 32,819 liquidations, where 9,191 (28.01 %) of these liquidations were performed by using Flashbots, 1,672 (5.09 %) by using flash loans, and 132 (0.4 %) by using both Flashbots and flash loans (see Table 1).

*3.1.4 Limitations.* Our measurements leverage heuristic-based MEV detection methods initially devised in prior research efforts [51, 53]. Our results, therefore, suffer from the same limitations as those works, and should be considered a lower bound on the number of MEV instances.

One such limitation in our methodology is that our sandwich MEV detection mechanism assumes that both sandwich transactions always occur within the same block. While this assumption is a boon to our ability to process the massive blockchain history efficiently—it is not strictly true. A single sandwich's transactions can be located on different blocks and still be successful (i.e., profitable for the submitter). Our methodology will not detect this.

Another limitation in our methodology is that we do not account for any other types of MEV. In this study we solely focus on the most well known and popular types: sandwiching, arbitrage, and liquidation. If other types exist, they would require their own unique detection methods and analyses.

Finally, we performed our measurements on popular exchanges and lending platforms. However, other exchanges and platforms may also suffer from MEV extraction and were not covered in this study simply because they are less popular.

## 3.2 Pending Transactions

We collected pending transaction via our archive node for a period of five months, between November 8th, 2021 and April 9th, 2022. We developed a script that used the standard Web3 API's [26] `web3.eth.subscribe` function to subscribe to "pendingTransactions" events, which are triggered upon receipt of incoming pending transactions, i.e, when new transactions enter the mempool. Next, our script retrieved the full corpus of details on a transaction via the Web3 API's `web3.eth.getTransaction` function and stored this information inside a MongoDB database. We collected a total of 125,660,856 public pending transactions.

## 3.3 Flashbots API

In line with their transparency initiative, Flashbots provides a public API [25], where users can download and inspect all bundles that have been mined by miners participating in the Flashbots network. The data is publicly accessible and contains information such as block number, miner address, miner reward, and transactions. Each transaction is labeled with a bundle ID and a bundle type.

We downloaded the entire list of Flashbots blocks until block 14,444,725 and obtained a list of 1,196,218 blocks that were mined as Flashbots blocks. We then used the transactions included in those blocks to identify and mark transactions as Flashbots transactions in our analysis.

## 3.4 Flash Loan Extraction

We identify flash loans by applying the techniques described by Wang et al. [54]. These techniques rely on crawling our archive node for specific events that are only triggered by lending platforms when a flash loan has been successfully executed. Our script is capable of detecting flash loans for the lending platforms Aave [7] and dYdX [6].

# 4 FLASHBOTS USAGE

The Flashbots project went live in January 2021, with the first Flashbots block being mined a few weeks later on February 11th, 2021 at block height 11,834,049. In this section, we analyze usage patterns since Flashbots' advent. In particular, we focus our analysis on four attributes of usage: (1) the number of Flashbots bundles, (2) a comparison of Flashbots usage to non-Flashbots usage, (3) participation and hashing power of Flashbots miners, and (4) usage towards MEV transactions.

## 4.1 Flashbots Bundles

The core unit of Flashbots usage is the *bundle*. According to the Flashbots protocol, all Flashbots transactions must end up on the blockchain wrapped in such a bundle. To understand how Flashbots is being used, we needed to analyze these bundles, which are published on the blockchain and consequently made available through Flashbots' public API.

In usage data collected from the public Flashbots API, we found a total of 3,249,003 bundles included in 1,196,218 blocks. The minimum number of bundles observed per Flashbots block is one and the maximum is 42. On average a Flashbots block contains 2.71 bundles, with a median of two bundles per block. Bundles contain on average 2.15 transactions, with a median of one. However, the

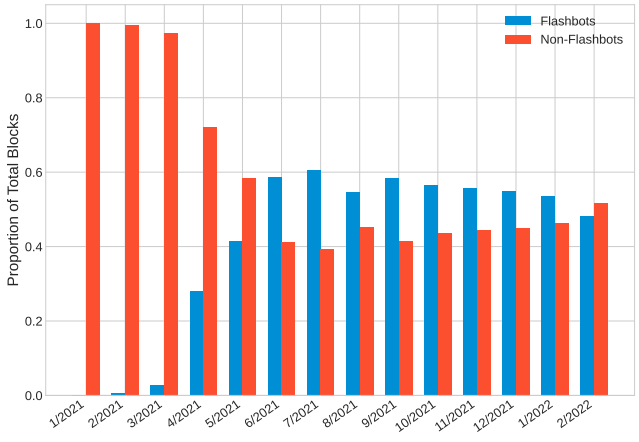



**Figure 3: Proportion of Flashbots blocks mined compared to *all* Ethereum blocks.**

largest bundle that we observed contained 700 transactions—this bundle was included in block 12,481,590. After further inspection, we found that the 700 transactions were miner payouts from F2Pool (one of the largest mining pools in Ethereum). 1,993,775 bundles or 61.37 % of the bundles only contain one transaction. These numbers suggest that there might not be many bundles available to mine, or we would expect to see more blocks with more bundles. The small number of bundles may indicate that either there are not many searchers submitting bundles, or their are not many MEV opportunities. We expand on these points in Section 4.5.

Not all bundle types (Section 2.5) are equally common. Only 1.9 % or 61,500 bundles were from *miner payouts*, and only 7.6 % or 247,523 bundles were marked as *rogue*. The rest of the bundles, 90.5 % or 2,939,980 were *flashbots* bundles; these include sandwiches, arbitrage MEVs, liquidation MEVs, and other order-dependent trades. These measurements suggest that the vast majority of Flashbots transactions are either submitted to extract MEV or to protect against MEV extractors.

## 4.2 Flashbots Blocks Vs. Non-Flashbots Blocks

One metric of Flashbots' popularity is the ratio of Ethereum blocks that include Flashbots transactions. This metric offers a high-level summary of Flashbots usage without factoring in the behaviors of individual participants. We calculate these ratios by summing the number of confirmed Flashbots blocks in each month and dividing by the total number of Ethereum blocks in that same month. We show these ratios in Figure 3. We see that adoption started low, as expected, but rose rapidly. The Flashbots ratio peaked in July 2021 at 60.6 %. It then hovered slightly above 50 % for several months. As of February 2022 the ratio dipped back below non-Flashbots blocks (48.2 %). As miners can be assumed to be rational, this decline indicates that fewer bundles are likely available.

This calculation is complicated by the fact that a Flashbots miner can decide arbitrarily to not include any Flashbots transaction. However, we do not believe this is likely to happen. The only two reasons that a Flashbots miner would not include any Flashbots transactions is because either none are available, or because they are ignoring them (or copying them for themselves). A rational miner will not ignore profitable transactions, and, we argue, miners will not copy searcher transactions, because the searchers make very little anyway (Section 5.1), and the miner would not want to risk getting kicked out of Flashbots, thus depriving them of future profit.

## 4.3 Hashing Power

The hashing power of Flashbots miners is a strong indicator of Flashbots' popularity among miners weighted by the contribution of those miners to the blockchain. Measuring the hashing power of miners directly is not possible without access to their hardware. We can, however, estimate a miner's hashing power (relative to the rest of the Ethereum network) by counting the number of blocks mined *by that miner* over a controlled time span and dividing by the total number of blocks in that same time span. This method yields the ratio of a miner's hashing power compared to the rest of the network.

As we only care about Flashbots miners, we do this analysis *only* for miners who have mined a Flashbots block in each time span examined (one month). We assume that if a miner is part of the Flashbots ecosystem (i.e., has mined at least one Flashbots block) then they contribute to the total Flashbots hashing power in that span, even if some of their blocks do not contain bundles. This is because a participating miner may not always mine Flashbots blocks, even if they are still be actively considering available bundles—bundles may not always be available, and some available bundles may not be sufficiently profitable to justify inclusion.

We present our findings in Figure 4. As expected, there were no Flashbots miners when the system went live in January 2021. Miners joined in quickly however, and by March accounted for 61.7 % of the Ethereum hashrate. By May the hash rate reached 97.6 %. As of February 2022, it was hovering around 99.9 %[2]. This means that effectively *all* prominent miners in Ethereum are enrolled in Flashbots. This has serious implications for the future of Ethereum as it has become a *de facto* part of the Ethereum protocol. Though initially jarring, the result should not be surprising. With the huge profit potential for miners at *no risk* (Section 2.5) and requiring only a small software change (in many cases, a 1-for-1 replacement if using `MEV-geth`), adoption is a rational decision. Also note that the network's hashing power is heavily skewed [36], so it is likely that far fewer than 99.9 % of miners have adopted Flashbots, but those who have are the ones with the most hashing power.

## 4.4 Participating Miners

We next discuss the number of miners participating in Flashbots and how their hashing power compares to each other. While the total hashing power of the network is useful, it is also valuable to know how much individual miners are contributing—this is especially true given Flashbots' calls for democratization (Goal 2). We count the number of miners who have mined at least $n$ blocks in each time span, where $n$ ranges logarithmically from $10^0$ to $10^4$. We present these results in Figure 5. This figure shows that the distribution of

[2]The Flashbots Transparency Dashboard [17] estimates a hashrate of 74.5 %, however, we suspect this to be an underestimate. They have not made their methodology for this measurement available.

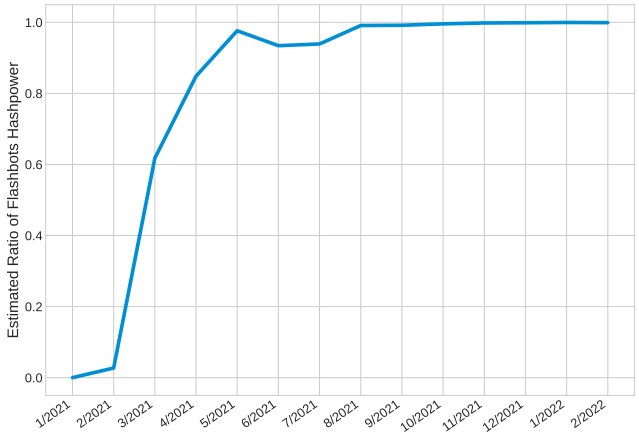



**Figure 4: Estimated hashrate of Flashbots miners as a fraction of total Ethereum hashpower.**

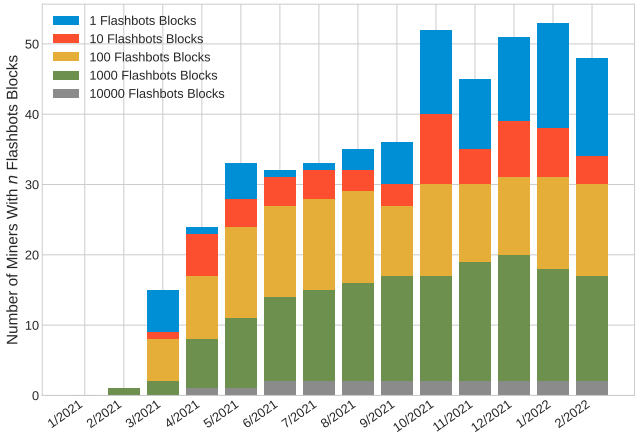



**Figure 5: The number of miners who have mined $n$ blocks in each month of our analysis.**

blocks mined (and therefore, hashing power) is long-tailed. This is consistent with previous findings [36]. One or two miners mined over 10,000 blocks (depending on the month), but at Flashbots's peak adoption, this is less than 2 % of the total number of Flashbots miners. Additionally, no month saw Flashbots blocks from more than 55 miners. In terms of Flashbots's goals, this result does not seem to indicate a democratization of MEV (Goal 2).

## 4.5 MEV Usage

Here we measure number of sandwiches for both Flashbots and non-Flashbots transactions over a 20-month period as well as the gas prices of these transactions. We present our findings in Figure 6. In the top plot, we notice an interesting anomaly. There is a steep and sudden drop in gas price in early April 2021. An astute reader might suspect this was caused by one of Ethereum's hard forks. We have plotted the time of the two nearest hard forks (code-named Berlin and London), but it is clear that they occurred well after the steep

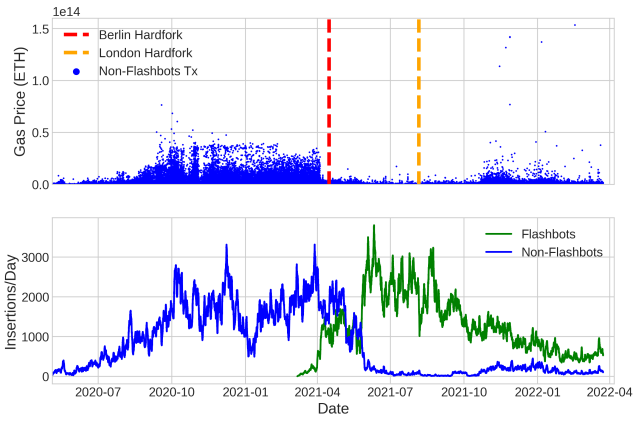


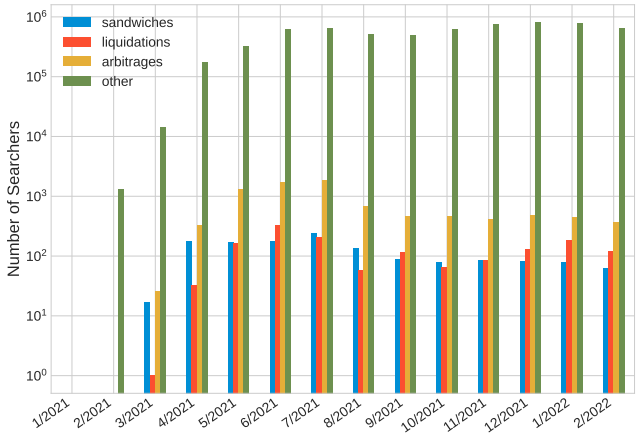

**Figure 6: Correlation between sandwich transactions per day and gas price.**

decline and well before the uptick seven months later. However, when plotted alongside the number of sandwiches (bottom plot in Figure 6), both Flashbots and non, we see that the precipitous drop correlates closely with the decline in Flashbots usage.

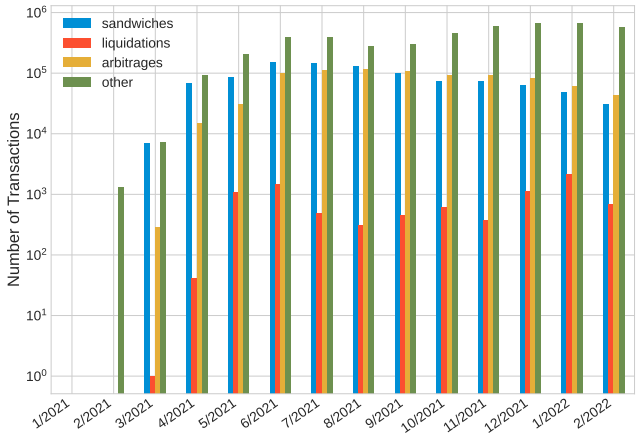
Interestingly, we notice that both Flashbots and non-Flashbots transactions reduce in frequency in September 2021. It is likely that this is due to the realization that Flashbots is not profitable for non-miners (Section 5.1) along with the concomitant rise in other private pools (Section 6). We hypothesize that the steep valley in Figure 6 may coincide with the popularity of Flashbots over time. The uptick in gas prices starting in September 2021 may be a result of decreased activity in Flashbots and simply be a return to pre-Flashbots behavior for non-miners.

*4.5.1 MEV Types.* Different types of MEV extractions will occur with different frequencies. In Figure 7 we track these frequencies over time. In the figure, we present the three types of MEV extractions as well as an *other* type, to include Flashbots transactions that are not MEVs—these are likely either order-dependent transactions or transactions that users wish to block MEV extractors from profiting off of. In Figure 7a, we show the number of Flashbots searchers that are engaging in each type of MEV. In all months, more searchers include *other* transactions than any of the MEV transactions (at least two orders of magnitude more). We also see an interesting pattern within each MEV type where they gradually increase through August 2021, before decreasing and leveling out. This suggests that after some initial buzz, many users left Flashbots for more profitable opportunities.

In Figure 7b, we see again that *other* transactions are the most popular type. However, the number of transactions of each type tend to be more consistent than the number of searchers. This indicates that the distribution may be heavily skewed with most MEV transactions are coming from small number of searchers. We also note that the number of sandwiches and arbitrage MEVs track each other closely, but there are far fewer liquidation transactions. This is likely because there are simply fewer liquidation opportunities.

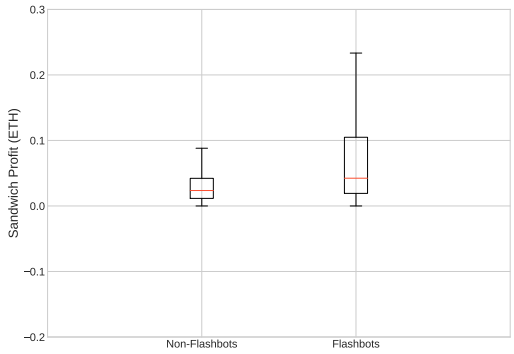



**(a) Searchers by MEV type.**

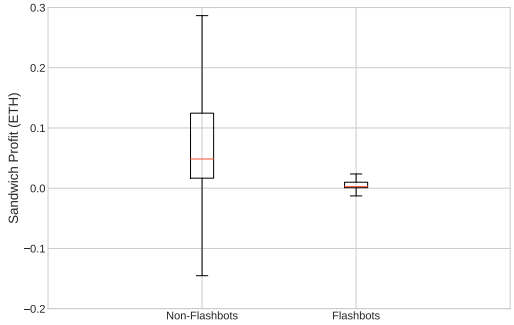



**(b) MEVs by type.**

**Figure 7: A breakdown of MEV transactions and searchers over town, separated by type, both y-axes are logarithmic.**




**(a) Miners**




**(b) Non-Miners (Searchers)**

**Figure 8: Sandwich profits (ETH) for different subpopulations. Horizontal bars represent the median profit.**

## 5 FLASHBOTS GOALS

The Flashbots project has established lofty goals (Benefit Distribution, Democratization, and Illumination). In this section, we evaluate how effectively it has achieved Benefit Distribution and Democratization. We discuss Illumination in Section 8.

### 5.1 Profit Distribution

Goal 3 of Flashbots is to distribute the benefits of MEV extraction. As such, Flashbots aims to give non-miners the opportunity to profit off of MEV. This is a reasonable proposition as MEV extracts value from *all* participants in the Ethereum ecosystem, so it makes sense to try to distribute profits among those members of the community. Concretely, this implies that miners, which are already flush with resources (e.g. money, powerful servers, etc.) should make a lower share of the profits as compared to non-miner participants (i.e., *searchers*)[3]. However, our measurements show that this is emphatically not the case.

We show, in Figure 8a, that miners are in fact seeing slightly higher average profits on sandwich frontrunning when using Flashbots than without Flashbots (0.125 ETH and 0.048 ETH, respectively). This comes at the cost of somewhat increased variance (standard deviation of 0.415 and 0.127, respectively). As the average profit is the *expected* profit, a rational miner will prefer to use Flashbots.

Conversely, non-miners (or in the Flashbots case: *searchers*) see notably lower average profit when using Flashbots: 0.02 ETH, compared to 0.13 ETH without Flashbots. Profit from non-Flashbots usage does, however, come with lower standard deviations, 0.154 versus 0.560. This means Flashbots is a more reliable income source—a important attribute for some users. However, this reliability is offset by lower expected income. Since miners are making 260 % more profit and non-miners are making 84.4 % less, it does not make rational sense for most non-miners to continue their sandwich MEVs in Flashbots.

In Figure 8b, we see that non-miners have significant probability of a sandwich MEV incurring a loss. We discuss this further in Section 5.2.

[3]This should not disincentivize miners, per se, as long as they make a higher cumulative profit.

## 5.2 Negative Profits

A number of transactions in the Flashbots epoch yielded negative profit for searchers which should not happen. A searcher standing to make negative profit would not forward such a bundle to a miner. Here we explore those transactions in depth.

In the Flashbots epoch, there have been 7,666 unprofitable MEVs out of a total of 485,680 transactions. About 1.58 %. Unprofitable transactions in Flashbots total 113.67 ETH or roughly 378,399.40 USD.

The cause of these negative profits is faulty contracts provided by the seachers in their submitted bundles. Integrating logical checks of correctness is difficult in any distributed system, no less for one as volatile as Ethereum. This is problematic in the context of the Flashbots ethos, as Flashbots is billed as a system to assist those without the resources to engineer their own involvement in MEV—it does not protect these low resource searchers from taking on losses due to unprofitable transactions. These occurrences do not well support Goal 2 (democratizing MEV extraction), and constitute a real risk for searchers.

# 6 PRIVATE MEV EXTRACTION

Flashbots is not the only private pool available to non-miners and miners. Other private pools have been proposed such as the Taichi Network [10] and the Eden Network [49]—the Eden Network is currently still active, but the Taichi Network is defunct as of October 15th, 2021 [10]. Along with well known private pools, miners can collaborate to make their own private pools, and even participate in multiple private pools concurrently. This raises the question of whether Flashbots is the most prominent private pool and whether miners only use Flashbots or if they also use other private pools concurrently.

Similar to Flashbots, both Taichi and Eden enable users to submit their private transactions to the network via a dedicated RPC endpoint, where a trusted miner will include these private transactions within a future block. Unlike Flashbots, however, neither network publicly discloses which private transactions were relayed via their network and mined by their miners. Currently known techniques are insufficient to identify how much MEV has been privately extracted using Taichi or Eden. We therefore only differentiate between public MEV extractions, Flashbots MEV extractions, and non-Flashbots private extractions MEV.

In this section, we investigate how much of the privately extracted MEV is due to Flashbots and how Flashbots contributes to the number of private transactions currently being mined in Ethereum.

## 6.1 Discovering Private MEV Extractions

The Ethereum blockchain does not include any explicit indication of whether a transaction was public or private. We next describe our method for inferring this information. At a high-level, we calculated the intersection between the set of publicly observed pending transactions and the set of transactions included in blocks on the blockchain—the transactions *not* in the intersection are, by definition, private. This assumes that our node saw the vast majority of transactions propagated through the network—an assumption consistent with previous measurements [41, 50].

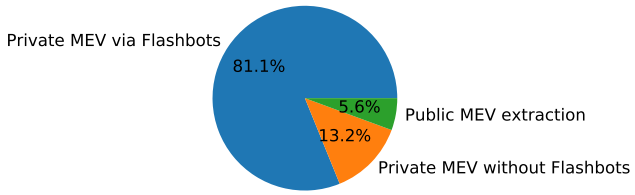



**Figure 9: Distribution of private vs. public MEV extraction.**

We did this intersection calculation for blocks ranging from block number 13,670,000 (i.e., November 23rd, 2021) to 14,444,725 (i.e., March 23rd, 2022). This range aligns with the window of our pending transactions data collection, and resulted in an analysis of 774,725 blocks—about 4 months of data.

Next, we identified the subset of private transactions which were MEV extractions. We did so by a similar technique as in Section 3.1.1. The difference is that we only searched for *MEVs* in the private transaction subset, while we only searched for *victims* in the public transaction set. This is because frontrunning other Flashbots transactions is disallowed in Flashbots and frontrunning other private pool transactions is not possible.

For example, sandwiches are always composed of three transactions: two transactions created by the MEV extractor, and one transaction created by the victim. The victim's transaction is ordered between the two transactions of the MEV extractor. Thus, to identify private sandwiches, we simply check whether both the first transaction and the third transaction are *not* part of our dataset of publicly observed pending transactions, while the second transaction *is* included in our dataset of publicly observed pending transactions.

## 6.2 Private MEV Distribution

Like Flashbots transactions and publicly propagated transactions, private transactions can be of any known type of MEV extraction. Analyzing the private transactions with this in mind, we found 80,093 (10.34 %) blocks containing at least one sandwich MEV within the 774,725 blocks we analyzed, In total, we found 99,928 sandwiches between November 23rd, 2021 and March 23rd, 2022. Out of these 99,928 sandwiches, 81,089 (81.15 %) were performed using Flashbots and the remaining 18,839 (18.85 %) were performed outside of Flashbots. Thus, the majority of the sandwich attacks between November 23rd, 2021 and March 23rd, 2022, were performed by using Flashbots.

When applying the heuristics described in Section 6.1 to detect private MEV extraction on sandwiches, we find that out of the 18,839 sandwiches that were not performed using Flashbots, 13,238 (70.27 %) were private. This means that only 5,601 sandwich attacks were public. In other words, from the overall 99,928 sandwich attacks only 5.6 % were carried out using the public mempool. This shows that private pools such as Flashbots are very popular for MEV extraction and that Flashbots is currently the dominating private pool. However, our results also show that other private pools seem to coexist and that miners engage in private MEV extraction.

### 6.3 Private Miner MEV Extraction

Miners and non-miners alike have flexibility in deciding which private pools, if any, they want to join. As we've argued, private pools can be lucrative in some cases, so here we evaluate the prevalence of miners and non-miners seeking such opportunities though non-Flashbots private pools. We identified miner addresses and account addresses that performed private non-Flashbots MEV sandwiches to better understand whether miners are engaging in MEV extraction themselves or whether they are part of broader private pools.

We found a total of 35 different miner addresses that mined private non-Flashbots MEV sandwiches and 41 account addresses that performed private non-Flashbots MEV sandwiches. We started by counting, for each account address, the number of unique miner addresses that mined an accounts address's private non-Flashbots sandwich MEV. The intuition here is that if a miner engaged in private MEV extraction then we should find an account address whose private non-Flashbots sandwich MEVs have only ever been mined by a single miner. It is very unlikely that all private MEV transactions would have been mined by the same miner within a private pool with multiple miners. Thus, if we find that an account address's private non-Flashbots sandwich MEV were mined by multiple miner addresses, then we can assume that these were mined via a private pool where these miners are participants.

We found two account addresses where each address's private transactions were only mined by a single miner, thus two miners that are likely engaging in private MEV extraction on their own. The first account address is `0x42B2C65dB7F9e3b6c26Bc6151CCf30CcE0fb99EA` with 30 private non-Flashbots sandwiches, all mined by the miner address `0x7F101fE45e6649A6fB8F3F8B43ed03D353f2B90c`, which according to Etherscan [21], is a part of Flexpool [23]. The second account address is `0xDD28D64E40e00aF54a0B5147539A515C4A0bC1c5` with 121 private non-Flashbots sandwiches, all mined by the miner address `0x829BD824B016326A401d083B33D092293333A830`, which again according to Etherscan, is part of F2Pool [22]. Since we did not see other account addresses engaging with only these two miners, we can eliminate the possibility that each of those two miners provide their own private pool for users. We also note that both miners have mined private transactions of other account addresses that have engaged with other miners. Thus, both Flexpool and F2Pool not only perform private MEV extraction on their own, but they also participate in other private pools besides Flashbots.

## 7 RELATED WORK

Eskandari et al. [34] were the first to propose a taxonomy of frontrunning attacks on the blockchain. However, Daian et al. [33] were the first to introduce the term "MEV" and to research frontrunning attacks from an economical perspective. They did so by monitoring the mempool and studying *priority gas auctions* (PGAs) in realtime. As a response to the negative effects of PGAs and the sharp increase of frontrunning attacks, projects such as Flashbots [13] and the Eden Network [49] have been proposed. These projects form private transaction pools which reflect private agreements with miners that allows users to bypass the public mempool and submit transactions privately.

While Eskandari et al. [34] provide a taxonomy and Daian et al. [33] show the existence of frontrunning in the wild, Torres et al. [53] were the first to quantify past frontrunning attacks using historical blockchain data. Qin et al. [51] on the other hand focused on quantifying MEV extraction. Both works discover large increases in frontrunning attacks over recent years. Our work leverages and extends their techniques in order to measure how much MEV is being extracted using private pools such as Flashbots. Qin et al. [51] also provide a theoretical analysis of network congestion in the presence of private pools, and conclude that Flashbots does not reduce the such congestion. While this may be true, our work shows that Flashbots does at least reduce gas prices in some instances.

Recent works by Piet et al. [50] and Capponi et al. [31] analyze the profit distribution within Flashbots and conclude, similarly to our work, that miners are earning most of the profit. However, neither work analyzes if this is unique to Flashbots or if this was already the case before Flashbots. Our work measures MEV extraction before and after the inception of Flashbots and concludes that searchers were making more profit prior to Flashbots. Our results also show that the number of searchers using Flashbots is decreasing. Moreover, while Piet et al. [50] focuses on measuring the distribution of private transactions, our work focuses on measuring the existence and distribution of private pools.

Numerous other frontrunning countermeasures have also been proposed. One strategy is to prevent frontrunning at the application layer either through an advanced commit-and-reveal scheme [30], tighter slippage protection [37], or a more frontrunning-resistant design of DEXes [27, 28, 32, 47, 57]. The main issue with these solutions is that they introduce new costs and each protects against only a single type of frontrunning attack. Another strategy is to prevent frontrunning at the consensus layer by proposing a fair ordering of transactions [39, 40, 43] or introducing transaction privacy [42, 56]. Unfortunately, none of these techniques are broadly adopted as they are not applicable to large public blockchains such as Ethereum.

## 8 CONCLUDING DISCUSSION

The rise of Flashbots indicates that many Ethereum users see value in what Flashbots offers. The public charter of Flashbots is laudable, but prior to this work, it was not clear if its goals were being met, and thus if it was a viable, long-term solution. This work yields the following three takeaways.

(1) Flashbots does not provide a one-stop, easy-access tool for MEV extraction. Users are still required to have the know-how to design MEV extraction solutions.
(2) Flashbots has increased the total number of MEV transactions in Ethereum, while reducing gas prices. Lower gas prices correspond to lower transaction fees for users.
(3) Flashbots has not been equally useful to all users. Miners have profited much more than searchers.

### 8.1 Mission Accomplished?

Despite Flashbots' efforts towards its goals, we must question how much progress it has actually made in achieving those goals. And if has made enough progress to be considered a success.

The first goal (Goal 1: Illuminating MEV) is to increase transparency of MEV in the mempool. If Flashbots has spurred more users to engage with other private pools then transparency cannot be said to be sufficiently addressed. It is not clear that, indeed, Flashbots was the driving factor in users moving towards other private pools, but private pools have certainly replaced public MEV to a large extent (Section 6.2).

Despite our efforts to contact the Flashbots project with the hope of measuring, directly, their pool of pending transactions, we were unable to get through to them. This is a strike against Flashbots as in this regard, it is no different than any other "dark", private pool. Conversely, their measurement dashboard [17] has many interesting plots and datapoints, though it contains no direct analysis of Flashbots' real world use cases. Another concern is that the methodology for generating these plots is not sufficiently open, making comparison difficult—if not impossible. Overall, there is no clear answer to whether Goal 1 has been achieved. Flashbots is much more open than other private pools, but does not seem to be an optimal solution in illuminating MEV behavior.

The second goal of Flashbots (Goal 2: Democratizing MEV) is similarly complex. Through Flashbots, more people have access to MEV-permitting infrastructure than ever before. However, this comes at the risk of taking on losses (Section 5.2), especially for non-miners—who already have fewer resources than their miner counterparts. This is because some degree of knowledge in writing smart contracts and understanding MEV opportunities is required to use Flashbots. This is beneficial for whoever can afford the effort to write (and verify) such contracts, but leaves non-experts in the same position as before. In fact, they are in a worse position, because now their are many more users who will be trying to frontrun them. Again, there is no clear answer to whether MEV is democratized, so we stick to the same refrain: more than before, but less than it could be.

The third goal of Flashbots is to distribute the benefits of MEV extraction (Goal 3: Distributing Benefits). On this point, we note that after an initially steep climb in usage, Flashbots has seen a decline in the number of blocks published (see Figure 3). If rational actors are declining in their usage of Flashbots, then it can only be assumed that they are not finding it as profitable as other options. This is unsurprising given that non-miners (a sizeable population) are seeing much less profit than they were before Flashbots (see Figure 8b). And so for those users no longer using Flashbots, the MEV problem has not been addressed. On this count, it is clear that Flashbots is not achieving its goal. Benefits are even more skewed than before.

## 8.2 Effects on Ethereum

MEV extraction does not happen in a vacuum. Their are externalities affecting the entire Ethereum ecosystem. On one hand, Flashbots has driven down gas prices (see Figure 6), which seems to be a good thing. We conjecture that the reduction in gas prices is because Flashbots logically splits the mempool into two disjoint pools, one for the MEV extractors and one for everyone else. Thus two different gas price auctions are occurring, the competition on one pool does not impact the other pool. This is contrary to what was previously happening where non-MEV extracting users had to pay more because of *priorty gas auctions* [33] happening in the public mempool.

On the other hand, fees for MEV extractors, have risen drastically. Flashbots has made it difficult for searchers to optimize bids, because of its sealed bid auctions. Searchers do not see the bids of other searchers, and so to increase their chances of extracting MEV, they prefer to increase their miner tip in order to be selected by the miner. This design choice solely benefits miners and not searchers. Perhaps another type of auction would be better. This gas-price tradeoff has come out squarely in favor of miners who now indirectly force searchers to pay higher fees—a tragedy of the commons.

## 8.3 Are We Going in the Right Direction?

We now question whether or not the goals of Flashbots are good goals, per se. In traditional financial markets, frontrunning is considered predatory behavior [46]. Many agree that it is equally pernicious in the context of blockchain economies [11, 35]. However, unlike other works that attempt to disallow frontrunning through cryptographic means [12, 39, 40, 43], Flashbots actually makes MEV extraction easier. Their reasoning being that MEV extraction is going to happen anyway, so it might as well be extracted in a way that is more broadly beneficial.

This view has been called into question. Juels et al. [38] directly call Flashbots, and other frontrunning-as-a-service (FaaS) solutions, "theft". Felten [35] describes Flashbots-like solutions as "a poorly designed tax on users." Arguing that the additional income for searchers is coming at the cost of others in the Ethereum ecosystem (i.e., miners).

It is no surprise that so many miners have joined Flashbots. The "London" hard fork resulted in a large reduction of miners' revenue through block rewards [44]. Flashbots' auction system seems to be designed with miner revenue in mind and thus makes it perfect for miners to compensate for the lost revenue due to the hard fork.

As discussed in Section 7, a number of technical solutions have been proposed to mitigate or reduce MEV extraction via frontrunning. Piet et al. [50] suggest changing the Ethereum protocol to randomly order transactions—while this is certainly possible using a random seed derived from the previous block, it has other critical flaws. Consider a sandwich MEV, which requires three transactions to appear in order. On average, after a random shuffle, the victim's transaction will appear in the middle of the block. Then, with 50 % probability, the first transaction in the sandwich will appear *before* the victim, and with 50 % probability, the third transaction in the sandwich will appear *after* the victim. This means, that despite the shuffle, there is still a 25 % chance that the MEV will take place. This probability doubles if we consider arbitrage or liquidation MEVs which only require a single frontrun or backrun. The probability of an MEV extractor could be further increased by simply including more transactions—essentially throwing darts, and hoping one sticks. This technique would certainly lower the likelihood of MEV success, but for cases with large enough payouts, the expected income would still be positive and thus the behavior is likely to continue. For this reason, we do not consider randomization viable.

Better approaches such as in [39, 40] offer cryptographic guarantees along with a definition of *fair-ordering* bespoke to the context

of blockchains. The limitation of these approaches being that they are currently only applicable to private blockchains where the number of nodes is small and reliable as opposed to the reality of public blockchains like Ethereum.

## 9 ETHICS

These measurements concern an actively used financial system. This system is operated by people who may take our analyses into account in their usage of (or abstention from) Flashbots and, more broadly, Ethereum. However, our work exclusively collected and parsed public data that was voluntarily distributed by the originators of that data. All data collected is in this work is considered publicly available by the protocol they operate within. Additionally, none of the analysis or collection focused on individuals, analysis was only done via aggregations.

## 10 ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers and our shepherd Michael Sirivianos for their valuable comments and feedback. This work was partly supported by the Ripple University Blockchain Research Initiative (UBRI) grant number 2018-188548 (5855).

## REFERENCES

[1] 2016. https://www.0x.org
[2] 2017. https://home.bancor.network
[3] 2018. https://uniswap.org
[4] 2019. https://compound.finance
[5] 2019. https://balancer.fi
[6] 2019. https://dydx.exchange
[7] 2020. https://aave.com/
[8] 2020. https://www.sushi.com
[9] 2020. https://curve.fi
[10] 2020. https://github.com/Taichi-Network/docs
[11] 2020. https://www.paradigm.xyz/2020/08/ethereum-is-a-dark-forest
[12] 2021. Fair Sequencing Services: Enabling a Provably Fair DeFi Ecosystem. https://blog.chain.link/chainlink-fair-sequencing-services-enabling-a-provably-fair-defi-ecosystem [Online; accessed 23. Nov. 2021].
[13] 2021. Overview | Flashbots Docs. https://docs.flashbots.net/flashbots-auction/overview [Online; accessed 23. Nov. 2021].
[14] 2022. https://github.com/flashbots/mev-geth
[15] 2022. https://github.com/flashbots/mev-inspect-rs
[16] 2022. https://github.com/ethereum/go-ethereum
[17] 2022. https://dashboard.flashbots.net/
[18] 2022. https://explore.flashbots.net/
[19] 2022. CoinGecko. https://www.coingecko.com/en/api [Online; accessed 18. May. 2022].
[20] 2022. CoinMarketCap. https://coinmarketcap.com/ [Online; accessed 18. May. 2022].
[21] 2022. Etherscan. https://etherscan.io [Online; accessed 18. May. 2022].
[22] 2022. f2pool. https://www.f2pool.com [Online; accessed 18. May. 2022].
[23] 2022. Flexpool.io. https://www.flexpool.io/ [Online; accessed 18. May. 2022].
[24] 2022. Live Updates: Ukraine Government Turns to Crypto to Crowdfund Millions of Dollars. https://www.elliptic.co/blog/live-updates-millions-in-crypto-crowdfunded-for-the-ukrainian-military [Online; accessed 02. September. 2022].
[25] 2022. mev-blocks. https://blocks.flashbots.net [Online; accessed 18. May. 2022].
[26] 2022. web3.js - Ethereum JavaScript API. https://web3js.readthedocs.io [Online; accessed 18. May. 2022].
[27] Carsten Baum, Bernardo David, and Tore Kasper Frederiksen. 2021. P2DEX: privacy-preserving decentralized cryptocurrency exchange. In *International Conference on Applied Cryptography and Network Security*. Springer, 163–194.
[28] Iddo Bentov, Yan Ji, Fan Zhang, Lorenz Breidenbach, Philip Daian, and Ari Juels. 2019. Tesseract: Real-time cryptocurrency exchange using trusted hardware. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1521–1538.
[29] Dan Bernhardt and Bart Taub. 2008. Front-running dynamics. *Journal of Economic Theory* 138, 1 (Jan 2008), 288–296. https://doi.org/10.1016/j.jet.2007.05.005
[30] Lorenz Breidenbach, Phil Daian, Florian Tramèr, and Ari Juels. 2018. Enter the Hydra: Towards Principled Bug Bounties and {Exploit-Resistant} Smart Contracts. In *27th USENIX Security Symposium (USENIX Security 18)*. 1335–1352.
[31] Agostino Capponi, Ruizhe Jia, and Ye Wang. 2022. The Evolution of Blockchain: from Lit to Dark. *arXiv preprint arXiv:2202.05779* (2022).
[32] Michele Ciampi, Muhammad Ishaq, Malik Magdon-Ismail, Rafail Ostrovsky, and Vassilis Zikas. 2021. FairMM: A fast and frontrunning-resistant crypto market-maker. *Cryptology ePrint Archive* (2021).
[33] Philip Daian, 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)*. IEEE, 910–927.
[34] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. 2019. SoK: Transparent Dishonesty: Front-Running Attacks on Blockchain. In *Financial Cryptography and Data Security - FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11599)*, Andrea Bracciali, Jeremy Clark, Federico Pintore, Peter B. Rønne, and Massimiliano Sala (Eds.). Springer, 170–189.
[35] Ed Felten. 2020. MEV auctions considered harmful. https://medium.com/offchainlabs/mev-auctions-considered-harmful-fa72f61a40ea
[36] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. 2018. *Decentralization in Bitcoin and Ethereum Networks*. Lecture Notes in Computer Science, Vol. 10957. Springer Berlin Heidelberg, Berlin, Heidelberg, 439–457. https://doi.org/10.1007/978-3-662-58387-6_24
[37] Lioba Heimbach and Roger Wattenhofer. 2022. Eliminating Sandwich Attacks with the Help of Game Theory. *arXiv preprint arXiv:2202.03762* (2022).
[38] Ari Juels, Ittay Eyal, and Mahimna Kelkar. 2021. Miners, Front-Running-as-a-Service Is Theft. *CoinDesk* (Apr 2021). https://www.coindesk.com/markets/2021/04/07/miners-front-running-as-a-service-is-theft
[39] Mahimna Kelkar, Soubhik Deb, Sishan Long, Ari Juels, and Sreeram Kannan. 2021. Themis: Fast, Strong Order-Fairness in Byzantine Consensus. *Cryptology ePrint Archive* (2021).
[40] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. 2020. Order-fairness for byzantine consensus. In *Annual International Cryptology Conference*. Springer, 451–480.
[41] Lucianna Kiffer, Asad Salman, Dave Levin, Alan Mislove, and Cristina Nita-Rotaru. 2021. Under the Hood of the Ethereum Gossip Protocol. 26.
[42] Eleftherios Kokoris-Kogias, Enis Ceyhun Alp, Linus Gasser, Philipp Jovanovic, Ewa Syta, and Bryan Ford. 2018. CALYPSO: Private data management for decentralized ledgers. *Cryptology ePrint Archive* (2018).
[43] Klaus Kursawe. 2020. Wendy, the good little fairness widget: Achieving order fairness for blockchains. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*. 25–36.
[44] Stefanos Leonardos, Barnabé Monnot, Daniël Reijsbergen, Stratis Skoulakis, and Georgios Piliouras. 2021. Dynamical Analysis of the EIP-1559 Ethereum Fee Market. *arXiv:2102.10567 [cs, math]* (Jun 2021). http://arxiv.org/abs/2102.10567 arXiv: 2102.10567.
[45] Michael Lewis. 2014. *Flash Boys*. W.W. Norton & Company.
[46] Viktor Manahov. 2016. Front-Running Scalping Strategies and Market Manipulation: Why Does High-Frequency Trading Need Stricter Regulation? *Financial Review* 51, 3 (Aug 2016), 363–402. https://doi.org/10.1111/fire.12103
[47] Conor McMenamin, Vanesa Daza, and Matthias Fitzi. 2022. FairTraDEX: A Decentralised Exchange Preventing Value Extraction. *arXiv preprint arXiv:2202.06384* (2022).
[48] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008), 9.
[49] Chris Piatt, Jeffrey Quesnelle, and Caleb Sheridan. 2021. Eden Network. (2021). https://edennetwork.io/EDEN_Network___Whitepaper___2021_07.pdf
[50] Julien Piet, Jaiden Fairoze, and Nicholas Weaver. 2022. Extracting Godl [sic] from the Salt Mines: Ethereum Miners Extracting Value. *arXiv:2203.15930 [cs]* (Mar 2022). http://arxiv.org/abs/2203.15930 arXiv: 2203.15930.
[51] Kaihua Qin, Liyi Zhou, and Arthur Gervais. 2021. Quantifying Blockchain Extractable Value: How dark is the forest? *arXiv:2101.05511 [cs]* (Dec 2021). http://arxiv.org/abs/2101.05511 arXiv: 2101.05511.
[52] Nick Szabo. 1997. Formalizing and Securing Relationships on Public Networks. https://firstmonday.org/ojs/index.php/fm/article/download/548/469
[53] Christof Ferreira Torres, Ramiro Camino, and Radu State. 2021. Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain. In *USENIX Security Symposium, Virtual 11-13 August 2021*.
[54] Dabao Wang, Siwei Wu, Ziling Lin, Lei Wu, Xingliang Yuan, Yajin Zhou, Haoyu Wang, and Kui Ren. 2020. Towards understanding flash loan and its applications in defi ecosystem. *CoRR* abs/2010.12252 (2020).
[55] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–32.
[56] David Yakira, Avi Asayag, Gad Cohen, Ido Grayevsky, Maya Leshkowitz, Ori Rottenstreich, and Ronen Tamari. 2021. Helix: A fair blockchain consensus protocol resistant to ordering manipulation. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1584–1597.
[57] Liyi Zhou, Kaihua Qin, and Arthur Gervais. 2021. A2mm: Mitigating frontrunning, transaction reordering and consensus instability in decentralized exchanges.

*arXiv preprint arXiv:2106.07371* (2021).

[58] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc Viet Le, and Arthur Gervais. 2021. High-Frequency Trading on Decentralized On-Chain Exchanges. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*. IEEE, 428–445.