mathematics

MDPI

Article

A Simple Characterization of MEV

Nicola Dimitri

Department of Economics and Statistics, University of Siena, Piazza San Francesco 7, 53100 Siena, Italy; dimitri@unisi.it

Abstract

In the last few years, the notion of Maximal Extractable Value (MEV) from a transaction in Ethereum has received much attention. Value from a transaction can be extracted in different ways. Yet, despite the wide interest generated by the topic and the variety of extraction strategies adopted, so far there seem to be only very few contributions trying to formalize the notion of MEV. This would be useful to investigate its main features, for a more complete description and a deeper understanding of the phenomenon. Based on the properties of its fundamental elements, that we define in the work, this paper presents a simple characterization of MEV.

Keywords: Maximal Extractable Value; Ethereum; blockchain

MSC: 91G15

1. Introduction

In recent years, the notion of Maximal Extractable Value (MEV) in Ethereum blocks production has been extensively discussed and investigated [1–7]. There is now a consensus that value from transactions could be extracted through a variety of strategies [8,9]. Some of the best-known such strategies are *arbitrage*, *liquidation*, *and front/back-running sandwiching*. The diffusion and relevance of the phenomenon have generated the need for data availability to estimate its overall impact in monetary terms, as well as the performance of alternative strategies [10,11]. For example, EigenPhi is a platform offering data provision service on the extent and type of MEV attacks.

A frequent connotation of MEV refers to extracting value from transactions/smart contracts in Ethereum, besides transactions fees. This takes place by changing the order of transactions and/or including additional transactions, which has attracted much attention in both the operational and research spheres. These are the so-called *front/back-running and sandwiching* strategies, whose specificity has induced the entrance, in block building and block validation activities, of important new actors in the Ethereum ecosystem, such as Flashbots [9,12]. There have also been contributions discussing how the Ethereum protocol for block formation could be modified to prevent MEV attacks [13]. Moreover, the efficacy of MEV extraction techniques on Ethereum has also been tested in alternative blockchains, like Algorand [14], where the arrival time of the transactions decides the order of their inclusion in a block.

In the paper we shall focus on these strategies and, henceforth, our discussion on MEV will only refer to them, excluding *arbitrage* and *liquidation* from the analysis.

Despite the several contributions on MEV, to our knowledge, very few attempts have been made to try to provide a general characterization of it in a formal way.

check for updates

Academic Editor: J.E. Trinidad-Segovia

Received: 2 October 2025
Revised: 7 November 2025
Accepted: 17 November 2025
Published: 1 December 2025

Citation: Dimitri, N. A Simple Characterization of MEV. *Mathematics* **2025**, *13*, 3845. https://doi.org/10.3390/math13233845

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Mazorra et al. (2022) [3] do present a definition of MEV, but within an approach and focus that differs from ours. A precise identification of the relevant elements associated with MEV extraction, and their properties, seems to be particularly compelling to clarify the main features of MEV and, possibly, to provide decision-making insights to relevant platforms.

The paper attempts to contribute to filling this gap. To do so, the work is structured as follows. In the next section we define what, for us, are the fundamental elements of a MEV attack on a transaction. As we shall see, these are the associated payoff and the set of ordered transactions used to obtain the MEV. In the work we call these two elements, respectively, the MEV mapping and the ATTACK mapping.

We then discuss some of the main features of such mappings, which allow us to propose a characterization of the MEV notion. Moreover, we extend the analysis to the MEV of a set of transactions and briefly discuss the possibility of a MEV self-attack. Some final considerations conclude the paper.

2. The Fundamental Elements to Characterize One-Transaction MEV

In this section we introduce what, for us, are the fundamental elements of the MEV associated with a single transaction. Initially, we do not specify the identity of the transaction submitter and the value extractor, assuming that they are different actors/users. Later, when considering the possibility of a so-called "self-attack", identities will have to be specified.

Suppose T is the set of transactions in the Ethereum network at some point in time, waiting to be stored in a block. In what follows we consider only MEV attacks on single transactions $t \in T$. For this reason, our approach is static, as we do not consider the dynamics of MEV attacks when the set T changes because blocks of transactions are introduced into the blockchain and new transactions are implemented.

Then, each transaction $t \in T$ for each *single user*, induces the following set:

(a) *(ATTACK DOMAIN of $t \in T$) The domain of a MEV attack to $t \in T$ is defined as the set of transactions $T^{a(t)} = T \cup a(t)$, that can be selected by the user to extract value from $t \in T$. Hence $a(t)$, such that $T \cap a(t) = \emptyset$, is the set of transactions added to T by the user for attacking t.*

We assume that, in principle, each transaction in the network that is not yet stored in the blockchain may generate value that a user could try to obtain by attacking it. Moreover, we assume that $a(t)$ contains all the additional transactions the user considers relevant to attack t, so that any $t' \notin T \cup a(t)$ is irrelevant for the attack. If $a(t) = \emptyset$, it means that to extract value from t, the attacker needs to add no transaction to T.

Based on (a), it is possible to introduce the following notion:

(b) *(ATTACK SET of $t \in T$) The subset $A(t) \subseteq T^{a(t)}$ is defined as the set of transactions that are effectively selected by a user to extract value from t.*

Hence, in general, the attacking set of transactions may contain elements from T and $a(t)$. It is natural to assume that $a(t) \subseteq A(t)$ since, otherwise, it would be pointless to implement additional costly transactions that are not used for the attack. It is worth observing that, in principle, *any* subset of $T^{a(t)}$ containing $a(t)$ could be an attack set, which is indeed what we are going to assume below.

For the next definition we introduce an additional notion. Take any set X of transactions; then $O(X)$ indicates the set of orderings and permutations of X. This allows us to define the following mapping.

(c) *(ATTACK MAPPING of $t \in T$) The mapping $O : A(t) \to O(A(t))$ defines the set of orderings and permutations of the transactions in $A(t)$ that can be implemented to extract value from t . Assuming that $|A(t)|$ is the number of transactions contained in the set $A(t)$,*

then $o(A(t)) \in O(A(t))$ is an ordering of the elements of $A(t)$ and $|A(t)|!$ is the number of possible orderings.

Hence, the set of orderings $O(A(t))$ is the image, codomain, of the attack mapping.

The last definition suggests that, to implement an MEV *attack* on t, the transactions in $A(t)$ must be ordered in a specific way when inserted into a block. In principle, such ordering can follow any criterion, although in blockchain practice ordering is based on transaction fees or timing.

Definitions (a)–(c) introduced the fundamental notions that identify which transactions, and their possible orderings, are implemented to extract value from t. In what follows, we now introduce additional definitions to identify the payoff obtained through such attacks.

(d) (EXTRACTABLE VALUE mapping) The mapping $V(o(A(t)):O(A(t)) \to R$ identifies the value that $o(A(t))$ can extract from transaction t . This is defined as the monetary payoff that a user can obtain when attacking t with $o(A(t))$.

The above definition of an extractable value is intended to be net of the costs paid to implement the attack. This immediately leads us to the following:

(e) (LOCAL MEV mapping) The mapping $M(A(t)) : A(t) \to R$ is the local MEV associated with transaction t when using $A(t)$ to attack t, defined as follows:

$$M(A(t)) = max_{o(A(t)) \in O(A(t))} V(o(A(t)))$$

(f) (MEV mapping) The mapping $M(t) : T \to R$ is the MEV associated with each transaction t , defined as follows:

$$M(t) = max_{A(t) \subseteq T^{a(t)}} M(A(t))$$

Notice that if $|T|$ stands for the number of transactions in T, the possible number of attacking sets $A(t) \subseteq T^{a(t)}$, such that $a(t) \subseteq A(t)$ and $t \in T$, is $2^{|T|} - 1$, which is exponentially large in $|T|$.

(g) (USER MEV) For each user, the associated MEV, which we indicate by M, is defined as follows.

$$M = max_{t \in T} M(t)$$

Some comments are in order. Definitions (d)–(g) capture the following idea: a transaction t can, in principle, be attacked in several ways, that is, with different $A(t) \subseteq T^{a(t)}$ and different orderings of $A(t)$. Alternative attacks may extract different values from t. Hence, for a given attack set $A(t)$, $M(A(t))$ is the maximum among those extractable values. $M(A(t))$ can also be negative to allow for the possibility of a non-profitable attack. Therefore, we defined $M(A(t))$ as a local MEV because it depends on the attack set. The MEV associated with transaction t is $M(t)$, defined as the maximum across all attack sets. Finally, the MEV for a user is defined as the maximum MEV over all transactions $t \in T$.

Therefore, it is immediately possible to extend the notion to a *global* MEV by considering the largest MEV across all users.

The following simple example illustrates the above notions. Assume $T = \{t_1, t_2\}$ contains two transactions. Moreover, suppose a user intends to perform a sandwich attack on transaction t_1, with transactions $a(t_1) = \{t_3, t_4\}$, so that $T^{a(t_1)} = \{t_1, t_2, t_3, t_4\}$.

If $A(t_1) = \{t_1, t_3, t_4\}$, then the associated set of permutations is given as follows:

$$O(A(t_1) = \{\{t_1, t_3, t_4\}; \{t_1, t_4, t_3\}; \{t_4, t_1, t_3\};\ \{t_4, t_3, t_1\};\ \{t_3, t_1, t_4\};\ \{t_3, t_4, t_1\}\}$$

Additionally, assume

$$V(\{t_1, t_3, t_4\}) = V(\{t_1, t_4, t_3\}) = 0$$

while

$$V(\{t_4, t_3, t_1\}) = V(\{t_3, t_4, t_1\}) = 3;\ V(\{t_4, t_1, t_3\}) = 5 \text{ and } V(\{t_3, t_1, t_4\} = 10$$

which implies that $M(A(t_1)) = 10$, since the most successful attack on t_1 is the sandwich attack $\{t_3, t_1, t_4\}$. Assuming, as we shall see in the next Section, that t_1 is included in any attacking set, then t_1 could also be attacked by $B(t_1) = T^{a(t_1)}$, with $O(B(t_1))$ containing 24 permutations. Then $M(t_1)$ is the maximum between $M(A(t_1))$ and $M(B(t_1))$. Finally, an analogous reasoning can be conducted for t_2 and the user MEV is given by the largest value between $M(t_1)$ and $M(t_2)$.

Having introduced the fundamental elements to investigate the MEV, below we are going to discuss some of their features to provide an initial description of MEV.

3. A Characterization of MEV: Some Features of $\mathbf{M(t)}$ and $\mathbf{A(t)}$

For the first characterization of MEV, we present a few features of $M(A(t))$ and $A(t)$. As above, we shall also consider the possibility of attacking t with alternative attack sets $B(t)$, with $B(t) \subseteq T^{a(t)}$.

Since, as with $A(t)$, we assume $a(t) \subseteq B(t)$, it follows that $(A(t) - B(t)) \cup (B(t) - A(t)) \subseteq T$; that is, two alternative attack sets that differ only for transactions in T. Likewise, $V(o(B(t))$ is the value extracted from t when attacking with the set $o(B(t))$, an ordering of $B(t)$.

(1) $t \in A(t)$ *for all* $A(t) \subseteq T^{a(t)}$ *. That is, a transaction is included in all subsets of its attacking transactions. This seems to be a compelling feature for the MEV attacks that we consider.*

(2) *If* $t = A(t)$, *then* $M(A(t)) = 0$. *That is, if an attacking set of transactions is composed of only the attacked transaction, then there is no positive value to extract from t when attacking with* $A(t)$.

Since $a(t) \subseteq A(t)$, then $t = A(t)$ implies that $a(t) = \emptyset$, and it follows that any attacking set $A(t)$ such that $A(t) \subseteq T^{a(t)}$ satisfies $A(t) \subseteq T$.

The following third feature may hold, though not necessarily for all the attacking sets.

(3) *A set* $A(t)$ *may satisfy the following: if* $t' \in A(t)$, *then, for the user,* $M(A(t)) \geqq M(t')$.

That is, when the user considers attacking t with $A(t)$, the associated local MEV $M(A(t))$ is no smaller than the MEV associated with any transaction in the attacking set. When the above feature holds, it means that it is inconvenient for the attacker to attack an attacking transaction.

As a matter of fact, an immediate, though interesting, benchmark implication of feature 3 is as follows:

Proposition 1. *Suppose features 1–3 hold for all* $A(t) \subseteq T^{a(t)}$ *and all* $t \in T$; *then* $M(t) = M$ *for all* $t \in T$.

Proof. Consider $t' \in A(t)$. Then, from feature (3), we have $M(t) \geqq M(A(t)) \geqq M(t')$. Moreover, feature (1) implies $t \in A(t)$, and choosing $A(t') = A(t) \cap T$, it follows that $t \in A(t')$. Therefore, from feature (3) we also have $M(t') \geqq M(A(t')) \geqq M(t)$, and the result is proved. □

It is interesting to conclude this section by asking the following question.

If $A(t)$ is a set MEV attacking t, then can any $B(t)$, such that $A(t) \subset B(t) \subseteq T^{a(t)}$ induce at least the same local MEV level as $A(t)$? Namely, is $M(A(t)) \leq M(B(t))$? Intuition suggests that the answer is yes, however with some qualifications. Before discussing this issue, notice that since $a(t) \subseteq A(t)$, it is also $a(t) \subseteq B(t)$ so that $A(t) \cap T \subseteq B(t) \cap T$. That is, the transactions in $B(t) - A(t)$ are not added to T, which implies that they generate no additional transactions fees to the attacker, unlike the attacking transactions in $a(t)$.

To see the point, consider the following example. Suppose a user is implementing a *sandwich* attack on transaction t by considering $A(t) = \{t, t_1, t_2\}$, with its ordering $o(A(t)) = \{t_1, t, t_2\}$, and that $V(o(A(t)) = M(A(t))$.

That is, the attack with $o(A(t)) = \{t_1, t, t_2\}$ induces the local MEV and consists of placing transaction t_1 before t and transaction t_2 after t.

Suppose now $B(t) = \{t, t_1, t_2, t_3\}$, and that the ordering $o(B(t)) = \{t_1, t, t_2, t_3\}$ is implemented. Then it is plausible that $V(o(B(t)) = V(o(A(t)) = M(A(t))$, as with the above ordering $o(B(t))$ transaction t_3 that appears after the attack and does not affect the ordering conditions before or during the attack by $o(A(t))$. That is, it would act like any other transaction stored in the same block.

Instead, the ordering $o'(B(t)) = \{t_1, t, t_3, t_2\}$ would interfere with the ordering $o(A(t)) = \{t_1, t, t_2\}$ and, possibly, be such that $V(o'(B(t)) \neq M(A(t))$. Therefore, the above considerations suggest that in general, $M(B(t)) \geq M(A(t))$.

The following fourth feature indicates how $M(B(t)) \geq M(A(t))$ may arise.

(4) A set $A(t)$ may satisfy the following. Define $A_C(t) = A(t) \cup C$, with $C \subseteq T$ and $A(t) \cap C = \emptyset$ Then, for some such C, it is that $M(A(t)) \leq M(A_C(t))$.

The above feature says that adding up to the attack set $A(t)$ and a set of transactions $C \subseteq T$ disjoint from $A(t)$ does not decrease the local MEV of $A(t)$.

Hence, the next result holds as follows:

Proposition 2. *Suppose feature 4 holds for all $A_C(t)$. Then, for any $B(t)$ such that $A(t) \subset B(t) \subseteq T^{a(t)}$, it holds that $M(A(t)) \leq M(B(t))$.*

Proof. Immediate. Define $C = B(t) - A(t) \subseteq T$ and the result follows. □

As above, notice that even though $A(t) \subset B(t)$, since $B(t) - A(t) \subseteq T$, launching the attack with $B(t)$ does not require any additional transaction fee to be paid by the attacker, because the added transactions are already in T.

Hence, a successful MEV attack on transaction t may be possibly implemented by means of several sets. Indeed, for any given $A(t)$, it is possible to define other successful attack sets by appropriately adding up and ordering transactions to $A(t)$.

Finally, it may be interesting to state the following simple result.

Proposition 3. *Suppose the attacking sets $A_i(t) \subseteq T^{a(t)}$, with $i = 1, 2 \ldots$, are the only ones such that $M(A_i(t)) = M(t)$. Moreover, assume there exists an index $i = k$ such that $A_k(t) \subseteq A_i(t)$ for all $A_i(t)$. Then $A_k(t)$ is the smallest set of transactions for a successful MEV attack on t.*

The last result identifies a *sufficient* set of transactions to extract MEV from transaction t. No transaction could be withdrawn from set $A_k(t)$ without compromising the success of the attack. It is intuitive to think that, in real life, $A_k(t)$ would represent the *most effective* set to obtain MEV, and thus the one likely to be chosen by the attacker.

4. The MEV of a Set of Transactions

In this section, we extend the notion of MEV to consider the possibility that a set of transactions $X \subseteq T$, appropriately ordered, rather than just a single transaction t, could be subject to an MEV attack. Though perhaps uncommon in real life, if at all, we believe that, in principle, it is interesting to consider such a generalization of MEV.

In analogy with the above notation, we use $M(X)$ to indicate the MEV associated with X, while $o(A(X))$ indicates an ordering of the attacking set of transactions $A(X) \subseteq T^{a(X)}$, where $T^{a(X)} = T \cup a(X)$ and $a(X)$, such that $a(X) \cap T = \emptyset$, stands for the additional set of transactions attacking X. Moreover, $M(A(X))$ is the local MEV obtained when attacking X with $A(X)$.

With sets of transactions, the following new feature can emerge.

(5) (*Monotonicity of MEV with respect to the attacked set*) *Two sets of transactions $X, Y \subseteq T$, where X and Y are appropriately ordered and such that $X \subseteq Y$, may satisfy $M(X) \leq M(Y)$. This is because $Y = X \cup (Y - X)$ and the attack on Y can always be focused on X only.*

To illustrate the point, consider the following example. Suppose $X = \{t_1, t_2\}$, $a(X) = \{t_3, t_4\}$, and that $o(A(X)) = \{t_3, t_1, t_2, t_4\}$ is implementing a *sandwich* MEV attack on X, such that $M(A(X)) = M(X)$. Moreover, consider $Y = \{t_1, t_2, t_5\} \subseteq T$, and suppose $a(X) = \{t_3, t_4\} = a(Y)$. Then, extending the previous reasoning for a single transaction t, it may be that the ordering $o(A(Y)) = \{t_3, t_1, t_2, t_4, t_5\}$ provides at least the same MEV as $o(A(X)) = \{t_3, t_1, t_2, t_4\}$, since in $o(A(Y))$ transaction t_5 appears after t_4.

It is important to note that for the monotonicity feature to hold, we assumed, as illustrated by the example, that in the ordering $o(A(Y))$ the additional transactions $a(X) = \{t_3, t_4\} = a(Y)$ could interfere with the initial ordering of $Y = \{t_1, t_2, t_5\}$, because in $o(A(Y))$ transaction t_4 is placed between t_2 and t_5. This seems to be a natural possibility to allow for, since in a block the order of the transactions stored could always be chosen by the validator.

An immediate implication of property (4) is that $M(t) \leq M(Y)$ may hold for $t \in Y$.

5. Self-Attack

In the previous sections we did not specify the identity of a transaction submitter and of the MEV attacker of that transaction. It was, however, assumed that if the submitter is user i, then the attacker is a different user j.

However, at least in principle, one could ask the following question: if i is the submitter of transaction t, can i also be the attacker of transaction t? That is, can there be an MEV self-attack? Although apparently paradoxical, in our framework the question seems to make perfect sense and deserving a discussion, albeit short.

In reality, self-attack does not seem to be widespread. Therefore, it would be interesting to understand why this is so. Among other possible reasons, we think there could be at least the following two. The first may be that i is unaware about MEV and, therefore, of the possibility of self-attack. The second is that i is aware of MEV, and also of the possibility of self-attack, which, however, is too costly to implement as compared to its returns.

An interesting example for this may be represented by swapping assets transactions in Automated Market Makers, such as Uniswap. Consider a liquidity pool with two assets, X and Y, that can be swapped with each other. Additionally, suppose the current quantity of the assets, respectively, x and y, are $x = 100 = y$. Finally, assume user i is interested in obtaining as many units of X as possible against 10 units of Y. In what follows we adopt the Uniswap Constant Product Function,

$$xy = k$$

to determine the exchange price between X and Y.

To do so we can envisage the following two procedures.

(a) The user swaps $\Delta y = 10$ against the quantity

$$\Delta x = \frac{x\Delta y}{y - \Delta y} = \frac{100(10)}{100 - 10} = 11$$

Hence, in this case, 1 unit of Y swaps with 1.1 units of X.

(b) Alternatively, the user can first swap $\Delta x = 11$ units of X to obtain in exchange $\Delta y = 10$ units of Y, so that now the liquidity pool is with $x = 111$ and $y = 90$. Then, the user can swap $\Delta y = 20$ units of Y against about $\Delta x = 32$ units of X. Hence, eventually, the pool size will be $x = 79$ and $y = 110$.

Therefore, subtracting from $\Delta x = 32$ the 11 units of X are initially deposited, the final number of X units obtained by the user is 21. Likewise, summing up $\Delta y = 10$ units of Y, obtained after the initial deposit of $\Delta x = 11$ units of X to the 10 units of Y available from the beginning, we reach the conclusion that with 10 units of Y, the user obtains 21 units of X. Therefore now 1 unit of Y provides 2.1 units of X, that is twice as much as before.

Procedure (b) can be interpreted as a *back-running* self-attack on the first transaction, which, from the example seems sufficiently simple and rewarding for the user. If this attack is not widely diffused, it could be because of the high transaction fees to be paid by the user or because the user may not have 11 units of X to swap. Moreover, the second attacking transaction must occur at the right time for the swapping price not to change too much. If this is not the case, the user may even lose rather than gain from the attack. Therefore, sufficiently risk-averse users may decide not to implement (b).

As a consequence, self-attacks may not be profitable for i.

6. Conclusions

In the paper we proposed a simple way to formalize the notion of MEV, concerning *front/back-running* and *sandwich* attacks, and, based on the defined fundamental elements, discussed some of its distinguishing features. We saw how the ATTACK mapping and the MEV mapping represent the building blocks of our framework, and their properties are key to providing a characterization of the MEV notion. Although in principle, non-negative value could be extracted from any transaction in the Ethereum network, we suggested that the MEV associated with the attacking transactions may be smaller than the MEV of the initially attacked transaction. This seems to be a needed requirement for the MEV mapping for attacks to enjoy some *stability*, namely to prevent users from beginning an attack and then dropping it to attack an attacking transaction. However, a thorough investigation of this topic would require a dynamic context, which we do not consider. We also argued that the same MEV could be obtained with alternative attack sets, although it is reasonable to think that users would select the smallest set of transactions to achieve their goal.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Daian, P.; Goldfeder, P.; Kell, T.; Li, Y.; Zhao, X.; Bentov, I.; Breidenbach, L.; Juels, A. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 910–927.

2. Obadia, A.; Salles, A.; Sankar, L.; Chitra, T.; Chellani, V.; Daian, P. Unity is Strength: A Formalization of Cross Domain Maximal Extractable Value. *arXiv* **2021**, arXiv:2112.01472. [CrossRef]
3. Mazorra, B.; Reynolds, M.; Daza, V. Price of MEV: Towards a Game Theoretical Approach to MEV. *arXiv* **2022**, arXiv:2208.13464v1. [CrossRef]
4. McMenamin, C. SOK: Cross Domain MEV. *arXiv* **2022**, arXiv:2308.04159. [CrossRef]
5. Babel, K.; Daian, P.; Kelkar, M.; Juels, A. Clockwork Finance: Automated Analysis of Economic Security in Smart Contracts. *arXiv* **2023**, arXiv:2109.04347v2. [CrossRef]
6. Mohan, V.; Khezr, P. Blockchains, MEV and the Knapsack Problem: A Primer. *arXiv* **2024**, arXiv:2403.19077v1. [CrossRef]
7. Guidi, B.; Michienzi, A. Linking MEV Attacks to Further Maximise Attackers' Gains: Evidence from the Ethereum Blockchain. *Blockchain Res. Appl.* 2025, *in press*. [CrossRef]
8. Barczentewicz, M. MEV on Ethereum: A Policy Analysis. *ICLE White Pap.* **2023**, 1–23. [CrossRef]
9. Yang, S.; Zhang, F.; Huang, K.; Chen, X.; Yang, Y.; Zhu, F. SoK: MEV Countermeasures: Theory and Practice. *arXiv* **2023**, arXiv:2212.05111. [CrossRef]
10. Weintraub, B.; Torres, C.; Nita-Rotaru, C.; State, R. A Flash(bot) in the Pan: Measuring Maximal Extractable Value in Private Pools. In Proceedings of the 22nd ACM Internet Measurement Conference, Nice, France, 25–27 October 2022; pp. 458–471.
11. Wunderlich, S. *Current State of MEV in the Ethereum System*; Hochschule Mittweida: Mittweida, Germany, 2023.
12. Bahrani, M.; Garimidi, P.; Roughgarden, T. Centralization in Block Building and Proposer-Builder Separation. *arXiv* **2024**, arXiv:2401.12120. [CrossRef]
13. Piet, J.; Nair, V.; Subramanian, S. MEVade: A MEV-Resistant Blockchain Design. In Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency, Dubai, United Arab Emirates, 1–5 May 2023.
14. Öz, B.; Rezabek, F.; Gebele, J.; Hoops, F.; Matthes, F. A Study of MEVExtraction Techniques on a First-Come-First-Served Blockchain. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, Avila, Spain, 8–12 April 2024; pp. 288–297.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.