Transaction Fee Mechanism Design*

Tim Roughgarden†

December 27, 2023

Abstract

Demand for blockchains such as Bitcoin and Ethereum is far larger than supply, necessitating a mechanism that selects a subset of transactions to include "on-chain" from the pool of all pending transactions. This paper investigates the problem of designing a blockchain transaction fee mechanism through the lens of mechanism design. We introduce two new forms of incentive-compatibility that capture some of the idiosyncrasies of the blockchain setting, one (MMIC) that protects against deviations by profit-maximizing miners and one (OCA-proofness) that protects against off-chain collusion between miners and users.

This study is immediately applicable to major change (made on August 5, 2021) to Ethereum's transaction fee mechanism, based on a proposal called "EIP-1559." Originally, Ethereum's transaction fee mechanism was a first-price (pay-as-bid) auction. EIP-1559 suggested making several tightly coupled changes, including the introduction of variable-size blocks, a history-dependent reserve price, and the burning of a significant portion of the transaction fees. We prove that this new mechanism earns an impressive report card: it satisfies the MMIC and OCA-proofness conditions, and is also dominant-strategy incentive compatible (DSIC) except when there is a sudden demand spike. We also introduce an alternative design, the "tipless mechanism," which offers an incomparable slate of incentive-compatibility guarantees—it is MMIC and DSIC, and OCA-proof unless in the midst of a demand spike.

1 Introduction

Real estate on a major blockchain is a scarce resource. For example, Bitcoin [39] and Ethereum [9], the two biggest blockchains by market cap[1], process roughly 5 and 15 trans-

*Date of initial journal submission: August 19, 2021. A four-page summary of this paper appeared in ACM SIGecom Exchanges, June 2021. A one-page abstract of this paper appeared in the Proceedings of the 22nd Annual ACM Conference on Economics and Computation, July 2021.

†Department of Computer Science, Columbia University. The author's research at Columbia University is supported in part by NSF awards CCF-2006737 and CNS-2212745. The author's work on a related report for the public [44] was supported by the Decentralization Foundation. Since January 1, 2022, the author has also served as Head of Research at a16z Crypto, a venture capital firm with investments in blockchain protocols. Email: tim.roughgarden@gmail.com.

[1] https://www.coingecko.com/

actions per second on average, respectively. Demand for these blockchains is far larger, necessitating a mechanism that selects a subset of transactions to include "on-chain" from the pool of all submitted transactions.

Historically, most blockchain protocols have employed a pay-as-bid transaction fee mechanism. Every transaction is submitted with a bid (in the blockchain's native currency), the miner of a block decides which transactions should be included in it, and upon publication of that block, the bid of each included transaction is transferred from its creator to the miner.[2] We follow blockchain convention and refer to this mechanism as a *first-price auction (FPA)*.

FPAs are natural enough and served for many years as the dominant paradigm in blockchain protocols, but are they really the best we can do? Could a different transaction fee mechanism offer more compelling incentive-compatibility properties?

The primary goal of this paper is to investigate these questions through the lens of mechanism design, while taking into account the many idiosyncrasies of the blockchain setting relative to more traditional applications of the field. For example:

(1) The miner of a block has dictatorial control over its contents, and in particular may deviate from the allocation rule intended by the protocol designer.

(2) The miner of a block can costlessly include fake transactions that are indistinguishable from real transactions.

(3) Payments should be computable from "on-chain" data, which typically discloses no information about losing bids.

(4) Miners and users can easily collude off-chain to manipulate a transaction fee mechanism.

The sequential and repeated nature of the blockchain setting also offers some potential advantages to the mechanism designer (which are not exploited by FPAs). For example:

(5) The choice of mechanism (such as a reserve price) for a given block could be informed by the (publicly visible) outcomes for previous blocks.

(6) Revenue from a block need not be transferred directly to the block's miner and could instead be redirected, for example to a foundation or to the miners of future blocks.

The key question is then: is there a transaction fee mechanism, possibly taking advantage of points (5) and (6), that meets the constraints imposed by (1)–(4) while also decreasing the strategic complexity relative to FPAs?

[2] "Miner" is the common term for block producers in a proof-of-work blockchain protocol (in which block validity rests on the inclusion of a partial pre-image of a cryptographic hash function). At the time the work described in this paper was done (in 2020–2021), Ethereum was a proof-of-work blockchain protocol. On September 15, 2022, in an event now known as "the Merge," the protocol upgraded to an alternative method of sybil-resistance called "proof-of-stake." In a proof-of-stake blockchain protocol, block producers are usually called "validators" rather than "miners." Because the model in this paper concerns the production of a single block and takes the identity of the block producer as fixed (agnostic to how it was selected), the Ethereum protocol's switch from proof-of-work to proof-of-stake does not change any of the incentive-compatibility results proved here.

EIP-1559

In addition to its basic scientific interest, the analysis of transaction fee mechanisms is immediately applicable to scrutinizing what is arguably the biggest change made to-date to the Ethereum blockchain. As background, a blockchain can change its own specification through a "hard fork," meaning a coordinated switch by the nodes in its network to a new and backwards-incompatible version of the protocol software. At the time of this writing, the Ethereum blockchain has had roughly fifteen hard forks since its genesis in May 2015. For example, the "London fork" began with block number 12865000, which was mined on August 5, 2021. Each hard fork implements a collection of Ethereum improvement protocols or "EIPs" that have been vetted by the community and approved for inclusion.

One of the EIPs implemented in the London fork is EIP-1559, a proposal by Vitalik Buterin (Ethereum's founder) [10, 12] that suggested several tightly coupled changes to Ethereum's transaction fee mechanism (which was previously an FPA), including the introduction of variable-size blocks, a history-dependent reserve price, and the burning of a significant portion of the transaction fees.[3] While Buterin did not provide a formal economic analysis of his proposed design, in [11] he outlined the motivation behind the proposal:

> Our goal is to discourage the development of complex miner strategies and complex transaction sender strategies in general, including both complex client-side calculations and economic modeling as well as various forms of collusion.

Community discussion of EIP-1559 began in earnest in early 2018[4] and over time the proposal garnered a number of advocates and critics. The polarization around this EIP was evident in the community survey conducted by Tim Beiko.[5] "Difficulties analyzing the EIP" ranked among the chief risks pointed out by the survey respondents, citing "the lack of a formal specification or proof for the mechanism that people can independently evaluate and critique." As part of the course of our work, we formalize the transaction fee mechanism proposed in EIP-1559 and rigorously interrogate to what extent it meets its stated design goals. Anecdotal evidence suggests that a preliminary report based on this work [44] contributed to the understanding of and broader support for EIP-1559.[6]

We stress that while EIP-1559 provides important and timely motivation for the present work, the discussion and results in this paper apply equally well to many other blockchains, including Bitcoin. The Bitcoin community is famously hostile to major changes to the Bitcoin protocol, however, so its transaction fee mechanism is likely to remain an FPA for the foreseeable future. Some smaller blockchains have deployed variations of the mechanism proposed in EIP-1559, including Filecoin[7] and NEAR[8].

[3] The results of the new transaction fee mechanism can be tracked at `https://ultrasound.money/`.

[4] `https://ethereum-magicians.org/t/eip-1559-fee-market-change-for-eth-1-0-chain/2783`

[5] `https://medium.com/ethereum-cat-herders/eip-1559-community-outreach-report-aa18be0666b5`

[6] See, for example, `https://thedefiant.io/eip-1559-user-dev-changes/`, `https://medium.com/centaur/centaur-library-how-eip-1559-will-lower-high-ethereum-gas-fees-a6a9b4f9583c` and `https://cryptonews.com/news/ethereum-s-hope-no-1559-what-it-does-and-what-it-doesn-t-do-11253.htm`.

[7] See `https://filfox.info/en/stats/gas`.

[8] See `https://near.org/papers/the-official-near-white-paper/`.

Paper Structure

Section 2 of this paper describes our basic model, a mechanism design setting in which the mechanism participants (creators of transactions) compete for transaction inclusion in a block with limited capacity. The formalism follows in the mechanism design tradition of allocation rules and payment rules, but with two twists. First, to take into account point (6) above, the payment rule (describing transfers to a block's miner) is supplemented with a burning rule (Definition 2.5) which indicates how much of the block's revenue is burned (or otherwise redirected away from the block's miner). Second, in light of point (3), payment and burning rules are restricted to depend only on the bids of the winning transactions.

Section 3 spells out three forms of incentive-compatibility, each guaranteeing robustness to a particular type of deviation from the intended behavior. Robustness to deviations from straightforward bidding by transaction creators is captured by the familiar notion of dominant-strategy incentive-compatibility (DSIC). The other two guarantees are specifically motivated by the blockchain setting and are new to this paper. First, we define incentive-compatibility for myopic miners (Definition 3.4), or MMIC, a condition stating that a transaction fee mechanism should be robust to deviations by profit-maximizing miners from the intended allocation rule (see point (1)) and also the injection of fake transactions (point (2)). Second, we define OCA-proofness (Definition 3.6), which states that the mechanism should be robust to cartels of transaction creators and miners colluding off-chain; more precisely, no off-chain arrangement among members of such a cartel should be capable of Pareto improving over a canonical on-chain outcome. The rest of the paper investigates to what extent different transaction fee mechanisms enjoy these three strategic robustness properties.

Section 4 provides a formal description of the transaction fee mechanism proposed in EIP-1559, which we call the 1559 mechanism. This mechanism makes use of a reserve price that is adjusted over time in response to excess demand, and variable-size blocks to provide an on-chain signal of excess demand. All revenue generated by the reserve price is burned. Finally, there is effectively an FPA sprinkled on top: Transaction creators also have the option of supplementing the reserve price by an additional "tip" that is transferred to the block's miner. Small tips should be sufficient to incentivize a miner to include a transaction during a period of stable demand, when there is room in the current block for all the outstanding transactions that are willing to pay the reserve price. Large tips can be used to encourage special treatment of a transaction, such as immediate inclusion in a block in the midst of a sudden demand spike. Section 4 also introduces the tipless mechanism, a variant of the 1559 mechanism in which the tips are hard-coded rather than user-specified. Relative to the 1559 mechanism, the tipless mechanism provides more robustness to non-straightforward bidding while sacrificing some resistance to off-chain collusion.

Section 5 contains the main results of this paper, and considers in turn the MMIC, DSIC, and OCA-proofness properties. Section 5.1 provides a sufficient condition for establishing the MMIC property (Theorem 5.2): the payment of a transaction should be independent of the other included transactions and their bids, and the allocation rule should maximize miner profit. This condition implies that FPAs, the 1559 mechanism, and the tipless mechanism are MMIC (Corollary 5.3). Second-price-type auctions are notable examples of non-MMIC mechanisms (Example 3.5).

Section 5.2 studies the extent to which different transaction fee mechanisms meet the DSIC condition. Theorem 5.5 shows that the tipless mechanism acts as a posted-price mechanism (with price equal to the reserve price plus the hard-coded tip) and, as such, is DSIC. The 1559 mechanism reverts to an FPA in the special case of a zero reserve price, but Theorem 5.7 identifies two conditions that together are sufficient for the optimality of straightforward bidding: the base fee should not be excessively low for the current demand curve (Definition 5.6), and transaction creators should use individually rational bidding strategies. Under these conditions, the 1559 mechanism acts as a posted-price mechanism, with price equal to the reserve price plus the miner's opportunity cost for transaction inclusion.

Section 5.3 considers OCA-proofness. Proposition 5.11 offers a characterization: A transaction fee mechanism is OCA-proof if and only if there is always an individually rational bidding strategy that leads to an outcome that maximizes the joint utility of the transaction creators and the block's miner. This proposition implies that FPAs and the 1559 mechanism are both OCA-proof (Corollaries 5.12 and 5.14, respectively). A key driver of the latter result is that the reserve price in the 1559 mechanism is determined solely by past history (leveraging the opportunity in point (5)) and not by a block's miner or its contents. Corollary 5.15 and Remark 5.16 show that the tipless mechanism loses OCA-proofness exactly in the regime in which the 1559 mechanism loses the DSIC property (that is, with an excessively low base fee). Finally, OCA-proofness considerations show that two of the major innovations in the 1559 mechanism, relative to an FPA—a reserve price and burned fees—are inextricably linked, as making either change without the other leads to a non-OCA-proof transaction fee mechanism. Intuitively, burning fees in an FPA catalyzes a miner and users to collude off-chain to avoid the burn (Corollary 5.17) while passing revenue from a reserve price to a block's miner opens the door for low-value transactions to pay the reserve price on-chain while receiving a partial refund from the miner off-chain (Corollary 5.18).

Section 6 concludes with a discussion of the two designs that fare best in our study (the 1559 and tipless mechanisms), a "pay-it-forward" alternative to money-burning, and the possibility of long-term collusion by cartels of miners.

Related Work

First-price auctions. Transaction fee mechanisms have been an integral part of blockchain protocol design since Nakamoto's original white paper introducing the Bitcoin protocol [39]. Since its genesis, Bitcoin has used an FPA as its transaction fee mechanism. The going price for Bitcoin transactions has been discussed much more thoroughly than the transaction fee mechanism itself. For example, the "blocksize wars" refers to a bitter dispute within the Bitcoin community over whether to increase the maximum allowable size of a block (ultimately leading in 2017 to a split between Bitcoin and a new fork called Bitcoin Cash) [7], and one of the primary arguments put forth by proponents of larger blocks was that they would prevent (or at least delay) high transaction fees that would be prohibitive for all but the most cost-insensitive participants. Another much-discussed issue, for example by Carlsten et al. [14] and Hasu, Prestwich, and Curtis [26], is whether Bitcoin becomes more vulnerable to various attacks as its block reward decreases (by a factor of 2 roughly every four years) and the transaction fees per block increase (as one would expect if the demand for Bitcoin transactions continues to increase).

Another line of work analyzes Bitcoin transaction fees as a market equilibrium. Houy [28] and Rizun [43] formalized the intuitive idea that equilibrium transaction fees should be determined by the matching of supply with demand. Richer models of demand, with waiting costs and pending transactions persisting until inclusion, are considered by Easley et al. [22] and Huberman et al. [29]. Among other results, these papers show that, as the fixed block reward decreases, the Bitcoin network can remain economically viable only if there is sufficient congestion (and consequent delays) to prop up the market-clearing transaction fees. More recently, Kiayias et al. [30] explore the related idea of using delays in a transaction fee mechanism to price differentiate between latency-sensitive and latency-insensitive transactions.

Alternative designs. Three previous works focus squarely on the design of alternative transaction fee mechanisms. Lavi et al. [32] and Yao [48], motivated by the aforementioned need to keep Bitcoin miner revenues high even as the block reward goes to zero, proposed the "monopolistic price" transaction fee mechanism. In this mechanism, all transactions included in a block pay the same amount (per unit size), which is the lowest bid by any included transaction. (See also Example 3.5.) Miners are then expected to maximize their revenue (price times quantity), which may involve restricting the supply (i.e., producing an underfull block) to prop up the price. Lavi et al. [32] and Yao [48] proved that this mechanism is "approximately DSIC," in the sense that truthful bidding is an approximately dominant strategy for users as the number of users grows large. With respect to the two new notions of incentive-compatibility introduced in this paper, the mechanism is MMIC but, on account of choosing revenue-maximization over joint utility-maximization, is not OCA-proof. Very recently, Nisan [41] studied the price dynamics of this mechanism over a sequence of blocks in a model with persistent transactions.

Basu et al. [6] also proposed an alternative transaction fee mechanism to FPAs, with an eye toward stronger incentive-compatibility guarantees; we summarize a slightly simplified version of it here. With the monopolistic price mechanism as a starting point, the mechanism in [6] adds two additional ideas. The first is to automatically charge only a nominal transaction fee to all transactions in any block that is not full. This rule is intended to prevent miners from boosting their revenue through the production of underfull blocks, though by itself the rule is toothless and leads to a mechanism equivalent to the monopolistic price mechanism (a miner can costlessly extend its favorite underfull block with minimum bid b to a full block with minimum bid b using fake transactions, all with bid b). The second new addition is that transaction fees are partially paid forward, with the transaction fee revenue from a block B split evenly between B's miner and the miners of the $\ell - 1$ subsequent blocks (here ℓ is a tunable parameter). Thus, the miner of a block gets a $1/\ell$ fraction of the transaction fee revenue in that block, along with a $1/\ell$ fraction of the combined revenue of the preceding $\ell - 1$ blocks. As a result, for $\ell \geq 2$, fake transactions now carry a cost: the miner pays their full transaction fees but recoups only a $1/\ell$ fraction of them as revenue. In our terminology, Basu et al. [6] prove that their mechanism is approximately MMIC and approximately DSIC provided the range of possible valuations is bounded and the number of transactions involved is sufficiently large. Perhaps the biggest vulnerability of this mechanism is its failure to satisfy OCA-proofness (for every $\ell \geq 2$): a miner and transaction creators could collude to move all transaction fees off-chain (paid to the miner), leaving no on-chain fees to pay forward to future miners (cf., Corollary 5.18).

Credible mechanisms. Our notion of incentive-compatibility for myopic miners (Definition 3.4) concerns an untrusted auctioneer (the miner), and as such is related to *credible mechanisms* [2]. Intuitively, a mechanism is credible if the agent tasked with carrying it out has no plausibly deniable utility-improving deviation. Interestingly, because miners can manipulate allocations but not prices (Remark 2.7), there is no need to restrict to "plausibly deniable" deviations in Definition 3.4. Another difference is that the current theory of credible mechanisms, and in particular the characterizations in [2], is largely restricted to single-item auctions (though see [20] for approximate revenue-optimality results in more general settings). Blockchain transaction fee mechanisms must work in the more general setting of (multi-item) knapsack auctions [1]. Finally, the model of credible mechanisms developed in [2] assumes that the auctioneer can communicate privately with each bidder, in sharp contrast to the publicly visible mempool of pending transactions and record of confirmed transactions that feature in the model here. Recent investigations of credible mechanisms in the context of blockchain protocols and decentralized finance include Ferreira and Parkes [24] and Chitra, Ferreira, and Kulkarni [16].

Recent related work. Many papers on transaction fee mechanism design (in addition to [16, 24, 30, 41], all cited above) have appeared since the publication of [44] and other preliminary versions of this work. Papers that are directly relevant to the specific mechanism proposed in EIP-1559 include Crapis, Moallemi, and Wang [18], Ferreira et al. [23], Leonardos et al. [33, 34], and Ndiaye [40], which investigate the properties of different base fee update rules; Liu et al. [36], Reijsbergen et al. [42], and Zhang and Zhang [49], which provide empirical analyses of the mechanism after its deployment; and Azouvi et al. [4] and Hougaard and Pourpouneh [27], which investigate multi-block strategies by (non-myopic) miners. More distantly related are the works of Canidio [13] and Kiayias, Lazos, and Schlegel [31], which explore the pros and cons of burning transaction fees (as in EIP-1559).

A number of recent papers study the design of transaction fee mechanisms more generally. Chung and Shi [17] introduce a collusion-resistance condition incomparable to OCA-proofness, which they call "c-side-contract-proofness (c-SCP)," and prove that no transaction fee mechanism can satisfy both the DSIC and the c-SCP conditions (for any positive integer c). The c-SCP condition requires that there is never a way for a miner and at most c users to collude through an OCA and increase their joint surplus (relative to the outcome in which those users bid truthfully and the miner executes the intended allocation rule). OCA-proofness, by contrast, only concerns OCA-enabled deviations by the "grand coalition," meaning the miner and *all* the users with transactions in the mempool. Intuitively, OCA-proofness protects against the miner and users banding together to "cheat the protocol" (cf., Corollaries 5.17 and 5.18); the c-SCP condition also protects against a subset of users joining forces with the miner to cheat the other users. See Gafni and Yaish [25] for further discussion of this point. Chung and Shi [17] show that their impossibility result can be partially circumvented if users have what they call "γ-strict" utility functions (intuitively, with each user assuming that overbidding will lead to comeuppance in the future); see also Tang and Yao [46].

The proof of the impossibility result in [17] also shows that, in the "infinite block size regime" in which the entire mempool would fit in a single block, every DSIC and 1-SCP transaction fee mechanism must burn all user payments (with the miner earning zero revenue). Several recent works have proposed changes to the model with the goal of evading this negative result: Shi, Chung, and Wu [45] propose using a multi-party computation to generate unmanipulatable randomness; Wu, Shi, and Chung [47] assume a lower bound on the number of non-strategic bidders; and Gafni and Yaish [25] restrict attention to a discrete valuation space.

Chen et al. [15], Gafni and Yaish [25], and Liu et al. [35] explore a range of design questions for non-DSIC transaction fee mechanisms, with an emphasis on Bayesian incentive-compatible mechanisms.

Finally, Bahrani, Garimidi, and Roughgarden [5] extend the model in this paper by allowing a miner to have its own private valuation for a block (as opposed to caring only about transaction fees and other in-protocol rewards). The motivation for this extension is "MEV," or "maximal extractable value" [19], which refers to the fact that a miner may be able to benefit, in a way invisible to the blockchain protocol, by including or excluding certain transactions. MEV has become particularly prominent in the Ethereum ecosystem since the rise of decentralized finance ("DeFi"), for example with miners frontrunning trades on decentralized exchanges. The main impossibility result in [5] shows that the presence of MEV makes transaction fee mechanism design significantly harder: even after setting aside any collusion-resistance requirements, with private miner valuations, no transaction fee mechanism can be incentive-compatible simultaneously for users and for miners.

2 Preliminaries

2.1 Blockchain Transactions

We consider blockchain protocols that operate in the following way (as in Bitcoin and Ethereum, for example). The blockchain protocol maintains state (such as account balances) and carries out an ordered sequence of transactions that read from and write to the current state (such as transfers of the blockchain's native cryptocurrency). We assume that each transaction t has a publicly visible and immutable *size* s_t, and that the creator of a transaction is responsible for specifying a *bid* b_t per unit size, indicating a willingness to pay of up to $s_t \cdot b_t$ (in the native currency) for the blockchain's execution of their transaction.[9]

A *block* is an ordered sequence of transactions and associated metadata (such as a reference to the predecessor block). There is a cap on the total size of the transactions included in a block, which we call the *maximum block size*.[10] Blocks are created and added to the blockchain by *miners*. Transactions are submitted by their creators to a peer-to-peer network; each miner monitors this network, maintains a *mempool* of outstanding transactions, and collects a subset of them into a block. To add a block to the blockchain, a miner provides a proof-of-work in the form of a solution to a computationally difficult cryptopuzzle; the puzzle difficulty is adjusted over time to maintain a target rate of block creation (in Ethereum, roughly one block per 13 seconds).[11] Importantly, the miner of a block has dictatorial control over which outstanding transactions are included and their ordering within the block. Transactions are considered confirmed once they are included in a block that is added to the blockchain. The current state of the blockchain protocol is then the result of executing all the confirmed transactions, in the specified order.[12]

The *transaction fee mechanism* is the part of the protocol that determines the amount that a creator of a confirmed transaction pays, and to whom that payment is directed. Historically, the biggest blockchains have used a separate first-price (i.e., pay-as-bid) auction for each block, with all proceeds going to the block's miner.

2.2 The Basic Model

This paper focuses primarily on incentives for miners and users at the time scale of a single block, and on several important types of attacks that can be carried out at this time scale (untruthful user bids, the insertion of fake transactions and other deviations by a miner, and off-chain agreements between miners and users). We discuss incentive issues and attacks that manifest over longer time scales in Section 6.

On the supply side, let C denote the maximum size of a block (C is for "capacity"). On the demand side, we use M to denote the set of transactions in a miner's mempool at the time of the current block's creation.

We associate three parameters with each transaction $t \in M$:

[9] For example, in Ethereum, transaction size is called the "gas limit" and is a measure of the cost (in computation, storage, and so on) imposed on the system by the transaction's execution. The most basic type of transaction (a simple currency transfer) requires 21,000 units of gas; more complex transactions require more gas.

[10] For example, in Ethereum, prior to the deployment of EIP-1559, the maximum block size was 15M gas, enough for roughly 600 of the simplest transactions.

[11] As noted in footnote 2, Ethereum switched from proof-of-work to proof-of-stake on September 15, 2022. In its proof-of-stake incarnation, miners are replaced by validators that have registered a specified amount of native currency into a designated staking contract. Every 12 seconds, the protocol chooses one validator uniformly at random (using pseudorandomness derived from the blockchain's state) as the one responsible for assembling the next block.

[12] Technically, a fork selection rule (e.g., longest-chain of a variant thereof) is used to resolve forks, meaning two or more blocks that claim a common predecessor. The confirmed transactions are then defined as those in the blocks that are well ensconced in the selected chain (that is, already extended by sufficiently many subsequent blocks).

- a *size* s_t;
- a *valuation* v_t per unit of size (in the native currency);
- a *bid* b_t per unit of size (in the native currency).

The valuation is the maximum per-size price the transaction's creator would be willing to pay for its execution in the current block. The bid corresponds to the per-size price that the creator actually offers to pay, which in general can be less (or more) than the valuation. The size and bid of a confirmed transaction are recorded on-chain; the valuation of a transaction is private to its creator.

2.3 The Design Space: Allocation, Payment, and Burning Rules

A transaction fee mechanism decides which transactions should be included in the current block, how much the creators of those transactions must pay, and to whom their payments are directed. These decisions are formalized by three functions: an *allocation rule*, a *payment rule*, and a *burning rule*. There are two significant differences between the formalism in this section and that in classical mechanism design, both dictated by blockchain idiosyncrasies: payments should depend only on on-chain information (see Remark 2.8), and revenue can be directed wherever the protocol sees fit (see Definition 2.5).

2.3.1 Allocation Rules

We use $\mathbf{H} = B_1, B_2, \ldots, B_{k-1}$ to denote the sequence of blocks in the current chain (with B_1 the initial genesis block and B_{k-1} the most recently added block) and M the pending transactions in the mempool. (Here $\mathbf{H}$ is for "history.")

Definition 2.1 (Allocation Rule) An *allocation rule* is a vector-valued function $\mathbf{x}$ from the on-chain history $\mathbf{H}$ and mempool M to a 0-1 value $x_t(\mathbf{H}, M)$ for each pending transaction $t \in M$.

A value of 1 for $x_t(\mathbf{H}, M)$ indicates transaction t's inclusion in the current block B_k; a value of 0 indicates its exclusion. We sometimes write $B_k = \mathbf{x}(\mathbf{H}, M)$, with the understanding that B_k is the set of transactions t for which $x_t(\mathbf{H}, M) = 1$.

We consider only feasible allocation rules, meaning allocation rules that respect the maximum block size C.

Definition 2.2 (Feasible Allocation Rule) An allocation rule $\mathbf{x}$ is *feasible* if, for every possible history $\mathbf{H}$ and mempool M,

$$\sum_{t \in M} s_t \cdot x_t(\mathbf{H}, M) \leq C. \qquad (1)$$

We call a set T of transactions *feasible* if they can all be packed in a single block: $\sum_{t \in T} s_t \leq C$.

Remark 2.3 (Miners Control Allocations) While a transaction fee mechanism is generally designed with a specific allocation rule in mind, it is important to remember that a miner ultimately has dictatorial control over the block it creates.

2.3.2 Payment and Burning Rules

The payment rule specifies the revenue earned by the miner from included transactions.

Definition 2.4 (Payment Rule) A *payment rule* is a function $\mathbf{p}$ from the current on-chain history $\mathbf{H}$ and transactions B_k included in the current block to a nonnegative number $p_t(\mathbf{H}, B_k)$ for each included transaction $t \in B_k$.

The value of $p_t(\mathbf{H}, B_k)$ indicates the payment from the creator of an included transaction $t \in B_k$ to the miner of the block B_k (in the native currency, per unit of size).

Finally, the burning rule specifies the amount of money burned for each of the included transactions.

Definition 2.5 (Burning Rule) A *burning rule* is a function $\mathbf{q}$ from the current on-chain history $\mathbf{H}$ and transactions B_k included in the current block to a nonnegative number $q_t(\mathbf{H}, B_k)$ for each included transaction $t \in B_k$.

The value of $q_t(\mathbf{H}, B_k)$ indicates the amount of money burned by the creator of an included transaction $t \in B_k$ (in the native currency, per unit of size). Burning money can be equated with a lump-sum refund to holders of the currency through deflation, à la stock buybacks. An alternative to money-burning that has similar game-theoretic properties is to redirect a block's revenue to entities other than the block's miner, such as a foundation or the miners of future blocks (see Section 6 for further discussion).

Example 2.6 (First-Price Auction) The (intended) allocation rule $\mathbf{x}^f$ in the first-price auctions historically deployed in Bitcoin and Ethereum is to include a feasible subset of outstanding transactions that maximizes the sum of the size-weighted bids. That is, the x_t^f's are assigned 0-1 values to maximize

$$\sum_{t \in M} x_t^f(\mathbf{H}, M) \cdot b_t \cdot s_t, \tag{2}$$

subject to (1).[13] A winner then pays its bid (per unit of size), with all revenue going to the miner (and none burned), no matter what the blockchain history and other included transactions: $p_t^f(\mathbf{H}, B_k) = b_t$ and $q_t^f(\mathbf{H}, B_k) = 0$ for all $\mathbf{H}$ and $t \in B_k$.

[13] In practice, some miners prefer to employ a greedy heuristic (ordering transactions by bid and including the largest feasible prefix of transactions) rather than solve this knapsack problem optimally. Because a typical block contains hundreds of transactions, the difference in revenue between a greedy and an optimal knapsack solution is usually negligible and can be safely glossed over.

Remark 2.7 (The Protocol Controls Payments and Burns) A miner does not control the payment or burning rule, except inasmuch as it controls the allocation, meaning the transactions included in B_k. Given a choice of allocation, the on-chain payments and fee burns are completely specified by the protocol. (Miners might seek out off-chain payments, however; see Section 3.3.)

Remark 2.8 (Mempool-Dependence) The allocation rule $\mathbf{x}$ depends on the mempool M because a miner can base its allocation decision on the entire set of outstanding transactions. Payment and burning rules must be computable from the on-chain history $\mathbf{H}$, and in particular cannot depend on outstanding transactions of M excluded from the current block B_k.[14]

Definition 2.9 (Transaction Fee Mechanism (TFM)) A *transaction fee mechanism* (or *TFM*) is a triple $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ in which $\mathbf{x}$ is a feasible allocation rule, $\mathbf{p}$ is a payment rule, and $\mathbf{q}$ is a burning rule.

A TFM is a mechanism for allocating transactions to a single block. A blockchain protocol is free to use different TFMs for different blocks, perhaps informed by the contents of previous blocks.

3 Miners, Users, and Incentive Compatibility

In a permissionless blockchain protocol such as Bitcoin or Ethereum, a mechanism designer must guard against harmful deviations from intended behavior by users (the creators of transactions), by miners, and by cartels of users and miners.

3.1 Users

We consider a notion of incentive compatibility for users that is familiar from traditional mechanism design, namely dominant-strategy incentive compatibility. Recall from Section 2.2 that the valuation v_t of a transaction t is the maximum price (per unit of size) the transaction's creator would be willing to pay for its inclusion in the current block. We assume that a user bids in order to maximize their net gain (i.e., the value for inclusion minus the cost for inclusion). To reason about the different possible bids that could be attached to a transaction t submitted to a mempool M, we use $M(b_t)$ to denote the result of adding the transaction t with bid b_t to M. For simplicity, we assume that each transaction in the current mempool has a distinct creator.[15]

[14] In principle, the state of a blockchain protocol could keep track of additional data useful for computing a payment or burning rule, such as the highest bid by an excluded transaction; see Chung and Shi [17] for an in-depth exploration of this idea.

[15] See Section 6 for further discussion.

Definition 3.1 (User Utility Function) For a TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$, on-chain history $\mathbf{H}$, and mempool M, the utility of the originator of a transaction $t \notin M$ with valuation v_t and bid b_t is

$$u_t(b_t) := \left(v_t - \underbrace{p_t(\mathbf{H}, B_k)}_{\substack{\text{payment to miner} \\ \text{(per unit size)}}} - \underbrace{q_t(\mathbf{H}, B_k)}_{\substack{\text{burn} \\ \text{(per unit size)}}} \right) \cdot s_t \tag{3}$$

if t is included in $B_k = \mathbf{x}(\mathbf{H}, M(b_t))$ and 0 otherwise.

In (3), we highlight the dependence of the utility function on the argument that is directly under a user's control, the bid b_t submitted with the transaction. Because our focus is on incentive issues on the single-block time scale, we do not explicitly model intertemporal effects such as waiting costs or otherwise provide additional foundations for the valuation v_t of immediate inclusion.

We assume that a transaction creator bids to maximize the utility function in (3). A TFM is then *dominant-strategy incentive compatible (DSIC)* if, assuming that the miner carries out the intended allocation rule, every user (no matter what their valuation) has a dominant strategy—a bid that always maximizes the user's utility (3), no matter what the bids of the other users.[16] FPAs are, of course, not DSIC. Vickrey-Clarke-Groves (VCG) mechanisms are classical examples of DSIC mechanisms, with truthful bidding a dominant strategy; as Example 3.5 shows, however, these mechanisms are problematic in a blockchain context.

3.2 Myopic Miners

We next formalize incentive compatibility at the single-block time scale from the perspective of a miner—intuitively, that the miner is incentivized to implement the intended allocation rule.

We include in our model of miner utility a *marginal cost* (per unit size), denoted by μ. (The casual reader is encouraged to take $\mu = 0$ throughout the paper.) This parameter reflects the fact that every transaction included in a block potentially imposes a small marginal cost on that block's miner.[17] The parameter μ can be interpreted as the minimum price that a profit-maximizing miner would be willing to accept in exchange for transaction inclusion when the maximum block size is not a binding constraint. For simplicity, we assume that μ is the same for all miners and common knowledge among users.

Remark 3.2 (First-Price Auctions Revisited) The first-price auction in Example 2.6 is stated for the case of $\mu = 0$. More generally, the miner should be expected to maximize

[16] To describe some of the transaction fee mechanisms used in practice, it will be convenient to allow dominant strategies other than truthful bidding in the definition of DSIC. The revelation principle (e.g., [38]) can be used to convert any such DSIC mechanism into one in which truthful bidding is a dominant strategy.

[17] For example, in proof-of-work blockchain protocols, the probability that a mined block is orphaned from the main chain (i.e., the "uncle rate") appears to increase roughly linearly with the block size [21].

its revenue minus its costs and the "$b_t \cdot s_t$" term in (2) should be replaced by $(b_t - \mu) \cdot s_t$:

$$\sum_{t \in M} x_t^f(\mathbf{H}, M) \cdot (b_t - \mu) \cdot s_t. \tag{4}$$

In addition to choosing an allocation (Remark 2.3), we assume that miners can costlessly add any number of fake transactions to the mempool (with arbitrary sizes and bids). We call a miner *myopic* if its utility function is its net revenue from the current block (given the transactions and bids submitted by the users).[18]

Definition 3.3 (Myopic Miner Utility Function) For a TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$, on-chain history $\mathbf{H}$, mempool M, fake transactions F, and choice $B_k \subseteq M \cup F$ of included transactions (real and fake), the utility of a *myopic miner* is

$$u(F, B_k) := \underbrace{\sum_{t \in B_k \cap M} p_t(\mathbf{H}, B_k) \cdot s_t}_{\text{miner's revenue}} - \underbrace{\sum_{t \in B_k \cap F} q_t(\mathbf{H}, B_k) \cdot s_t}_{\text{fee burn for miner's fake transactions}} - \underbrace{\mu \sum_{t \in B_k} s_t}_{\text{marginal costs}} . \tag{5}$$

The first term sums over only the real included transactions, as for fake transactions the payment goes from the miner to itself. The second term sums over only the fake transactions, as for real transactions the burn is paid by their creators (not the miner). In (5), we highlight the dependence of the utility function on the two arguments that are under a miner's direct control, the choices of the fake transactions F and included (real and fake) transactions B_k.[19]

A transaction fee mechanism is generally designed with a specific allocation rule in mind (Remark 2.3), but will miners actually implement it?

Definition 3.4 (Incentive-Compatibility for Myopic Miners (MMIC))
A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is *incentive-compatible for myopic miners (MMIC)* if, for every on-chain history $\mathbf{H}$ and mempool M, a myopic miner maximizes its utility (5) by creating no fake transactions (i.e., setting $F = \emptyset$) and following the suggestion of the allocation rule $\mathbf{x}$ (i.e., setting $B_k = \mathbf{x}(\mathbf{H}, M)$).

For example, FPAs are MMIC—the intended allocation rule maximizes miner net revenue, which the miner is happy to do (see also Corollary 5.3). Second-price-type auctions are not MMIC, however, as in many cases a miner can boost its revenue through the inclusion of fake transactions:

Example 3.5 (Second-Price-Type Auctions Are Not MMIC) Consider a collection of transactions, all of unit size, and a block that has room for three of them. In this setting, a second-price-type auction would prescribe including the three transactions with the highest

[18] Miners may also receive transaction-independent rewards from a blockchain protocol for producing a block, such as the "block reward" in Bitcoin. Such rewards are independent of the miner's actions and therefore irrelevant for our game-theoretic analysis.

[19] We can assume that $F \subseteq B_k$, as there's no point to creating and then excluding a fake transaction.

bids and charging each of them the lowest of these three bids.[20] Now imagine that the top three bids are 10, 8, and 3. If a miner honestly executes the auction, its revenue will be $3 \times 3 = 9$. If the miner instead submits a fake transaction with bid 8 and executes the auction (with the top two real transactions included along with the fake transaction), its net revenue jumps to $2 \times 8 = 16$.

3.3 Off-Chain Agreements

Another idiosyncrasy of the blockchain setting is the easy availability of side channels and the consequent risk of off-chain collusion by users and miners. This danger is not hypothetical for a general smart contracts platform such as Ethereum, where off-chain markets are already common in practice.[21]

Definition 3.6 (Off-Chain Agreement (OCA)) For a miner m and a set T of transactions, an *off-chain agreement (OCA)* between T's creators and m specifies:

(i) a bid vector $\mathbf{b}$, with b_t indicating the bid to be submitted with the transaction $t \in T$;

(ii) an allocation vector $\mathbf{x}$, indicating the transactions that the miner m will include in its block;

(iii) a per-size transfer τ_t from the creator of each transaction $t \in T$ to the miner m. (If $\tau_t < 0$, the transfer should be interpreted as a refund from the miner to the transaction creator.)

In an OCA, each creator of a transaction t agrees to submit t with an on-chain bid of b_t while transferring $\tau_t \cdot s_t$ to the miner m off-chain; the miner, in turn, agrees to mine a block comprising the agreed-upon transactions of T.

Intuitively, we define a TFM to be "OCA-proof" if no OCA Pareto improves over a canonical on-chain outcome. More formally, by a *bidding strategy*, we mean a function $\sigma : \mathbb{R}^+ \to \mathbb{R}^+$ mapping user valuations to on-chain bids. For a valuation profile $\mathbf{v}$, $\sigma(\mathbf{v})$ denotes the bid vector obtained by the component-wise application of σ. With respect to a fixed TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$, a bidding strategy σ is *individually rational* if collective bidding according to σ guarantees nonnegative utility for all. Equivalently (using (3)), a bidding strategy is individually rational if for every on-chain history $\mathbf{H}$, transactions T with valuations $\mathbf{v}$, and transaction t in $B_k = \mathbf{x}(\mathbf{H}, T(\sigma(\mathbf{v})))$:[22]

$$p_t(\mathbf{H}, B_k) + q_t(\mathbf{H}, B_k) \leq v_t.$$

[20] A classical Vickrey auction would prescribe charging the highest losing bid rather than the lowest winning bid. The former is off-chain and thus unusable in a blockchain context, while the latter is on-chain and typically a close enough approximation.

[21] See, e.g., `https://docs.flashbots.net/flashbots-auction/overview/`.

[22] Here $T(\mathbf{b})$ denotes the mempool with transactions specified by T and bids specified by $\mathbf{b}$.

Definition 3.7 (OCA-Proof) A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is *OCA-proof* if, for every on-chain history $\mathbf{H}$, there exists an individually rational bidding strategy $\sigma_{\mathbf{H}}$ such that, for every possible set T of outstanding transactions and valuations $\mathbf{v}$, there is no OCA under which the utility (3) of every transaction creator and the utility (5) of the miner is strictly higher than in the outcome $B_k = \mathbf{x}(\mathbf{H}, M(\sigma_{\mathbf{H}}(\mathbf{v})))$ with on-chain bids $\sigma_{\mathbf{H}}(\mathbf{v})$ and no off-chain transfers.[23]

In other words, if a TFM is *not* OCA-proof, then there is a possible blockchain history such that, no matter what individually rational bidding strategy users use, there will be cases in which off-chain collusion collectively benefits the miner and users.

OCA-proofness can differentiate seemingly similar TFMs. For example, we'll see in Section 5.3 that first-price auctions in which all proceeds go to the miner are OCA-proof, while those in which any amount of revenue is burned are not (intuitively, because OCAs allow the miner and users to coordinate and evade the intended burn).

4 The 1559 and Tipless Mechanisms

This section formalizes the description of the transaction fee mechanism proposed in EIP-1559, along with an alternative design that offers a different set of trade-offs (a stronger incentive-compatibility guarantee for users but weaker resistance to off-chain agreements). Both will serve in the next section as running examples for our main results.

Burning a history-dependent base fee. Here are the first three (of eight) key ideas in EIP-1559:

1. Each block has a protocol-computed reserve price (per unit size) called the *base fee.* Paying the base fee is a prerequisite for inclusion in a block.

2. The base fee is a function of the preceding blocks only, and does not depend on the transactions included in the current block.

3. All revenues from the base fee are burned—that is, permanently removed from the circulating supply of the native currency.

The second point is underspecified; how, exactly, is the base fee derived from the preceding blocks? Intuitively, increases and decreases in demand should put upward and downward pressure on the base fee, respectively. But the blockchain records only the confirmed transactions, not the transactions that were priced out. If miners publish a sequence of maximum-size blocks, how can the protocol distinguish whether the current base fee is too low or exactly right?

[23] This definition subsumes and corrects the preliminary definition of OCA-proofness in [44].

Variable-size blocks. The next key idea is to relax the constraint that every block has size at most C and instead require only that the *average* block size is at most C. The mechanism in EIP-1559 then uses past block sizes as an on-chain measure of demand, with big blocks (size above C) and small blocks (size less than C) signaling positive and negative excess demand, respectively. Some finite maximum block size is still needed to control network congestion, which in EIP-1559 is twice the average block size:

4. Double the maximum block size (i.e., define $C_{max} := 2C$), with the old maximum C serving as the new *target* block size ($C_{target} := C$).

5. Adjust the base fee upward or downward whenever the size of the latest block is bigger or smaller than the target block size, respectively.[24]

If the base fee is burned rather than given to miners, why should miners bother to include any transactions in their blocks at all? Also, what happens when there are lots of transactions (with total size exceeding C_{max}) willing to pay the current base fee?

Tips. The transaction fee mechanism proposed in EIP-1559 addresses the preceding two questions by allowing the creator of a transaction to specify a *tip*, to be paid above and beyond the base fee, which is transferred to the miner of the block that includes the transaction (as in a first-price auction). Intuitively, small tips should be sufficient to incentivize a miner to include a transaction during a period of stable demand, when there is room in the current block for all the outstanding transactions that are willing to pay the base fee. Large tips can be used to encourage special treatment of a transaction, such as the immediate inclusion in a block in the midst of a sudden demand spike. The final ingredients of the mechanism in EIP-1559 are:

6. Rather than a single bid, the creator of a transaction is now responsible for specifying both a *tip* and a *fee cap* for it. A transaction will be included in a block only if its fee cap is at least the block's base fee.

7. If a size-s transaction with tip δ and fee cap c is included in a block with base fee r, the transaction creator pays a total of $s \cdot \min\{r + \delta, c\}$.

8. Revenue from the base fee (that is, $s \cdot r$) is burned and the remainder ($s \cdot \min\{\delta, c - r\}$) is transferred to the miner of the block.

Thus, with respect to a base fee r, a transaction t with tip δ_t and fee cap c_t is interpreted as a transaction with bid $b_t = \min\{r + \delta_t, c_t\}$.[25]

[24] Precisely, empty and maximum-size blocks decrease and increase the base fee by 12.5%, respectively, with the effect of other block sizes then determined by linear interpolation. (In particular, blocks matching the target size do not alter the base fee.)

[25] Our model and results consider only the single-block time scale, within which the base fee of this mechanism would be fixed and public. As such, the components of the mechanism that govern how the base fee evolves over multiple blocks, including the relationship between C_{target} and C_{max} and the exact formula for

The 1559 mechanism. We are now in a position to phrase EIP-1559's transaction fee mechanism—for shorthand, the *1559 mechanism*—in the language of Section 2. Because the base fee of a block depends solely on the contents of the preceding blocks, we can denote by $\alpha(\mathbf{H})$ the base fee of a block B_k with history $\mathbf{H}$.

Definition 4.1 (1559 Mechanism) For each history $\mathbf{H}$ and corresponding base fee $r = \alpha(\mathbf{H})$:

(a) the (intended) allocation rule $\mathbf{x}^*$ of the 1559 mechanism is to include a feasible subset of outstanding transactions that maximizes the sum of the size-weighted bids, less the cost and total base fee paid (and subject to the block size constraint (1), with capacity C_{max}):

$$\sum_{t \in M \,:\, b_t \geq r} x_t^*(\mathbf{H}, M) \cdot (b_t - r - \mu) \cdot s_t; \tag{6}$$

(b) the payment rule of the 1559 mechanism is

$$p_t^*(\mathbf{H}, B_k) = b_t - r$$

for all $t \in B_k$;

(c) the burning rule of the 1559 mechanism is

$$q_t^*(\mathbf{H}, B_k) = r$$

for all $t \in B_k$.

The tipless mechanism. We next define the *tipless mechanism*, so called because it is essentially the 1559 mechanism with constant and hard-coded tips rather than variable and user-specified tips. As with the 1559 mechanism, each block has a base fee $r = \alpha(\mathbf{H})$ that depends on past blocks and is burned. The creator of a transaction t specifies a fee cap c_t but no tip. This parameter induces a bid b_t for the transaction with respect to any given base fee r, namely $b_t = \min\{r + \delta, c_t\}$. Here δ is a hard-coded tip to incentivize miners to include transactions—for example, equal to (or perhaps slightly higher than) the marginal cost μ.[26] In effect, the bid space of the mechanism is $[0, r+\delta]$, with higher bids automatically interpreted as $r+\delta$ by the protocol. Only transactions with bid $r+\delta$ are eligible for inclusion in a block with base fee r; transactions with lower bids included in the block are considered invalid by the protocol.

the base fee update rule, are not directly relevant for our analysis. However, the interpretation of Theorem 5.7 and Corollary 5.15 as "usually DSIC" and "usually OCA-proof" guarantees does implicitly assume a base fee update rule that, under "normal conditions," prevents the base fee from becoming excessively low relative to the current demand (in the sense of Definition 5.6); see also footnote 31.

[26] More generally, the hard-coded tip δ could be adjusted over time via hard forks, as is typically done for a number of other protocol parameters.

Definition 4.2 (Tipless Mechanism) Fix a hard-coded user tip δ. For each history $\mathbf{H}$ and corresponding base fee $r = \alpha(\mathbf{H})$:

(a) the (intended) allocation rule $\mathbf{x}^\delta$ of the tipless mechanism is to maximize miner revenue from eligible transactions (i.e., those with bid at least $r + \delta$), less costs and subject to (1) with maximum block size C_{max}:

$$\sum_{t \in M \,:\, b_t \geq r+\delta} x_t^\delta(\mathbf{H}, M) \cdot (\delta - \mu) \cdot s_t, \tag{7}$$

or equivalently, in the intended regime with $\delta \geq \mu$, to include a largest-possible subset of eligible transactions;[27]

(b) the payment rule of the tipless mechanism is

$$p_t^\delta(\mathbf{H}, B_k) = \delta$$

for all $t \in B_k$;

(c) the burning rule of the tipless mechanism is

$$q_t^\delta(\mathbf{H}, B_k) = r$$

for all $t \in B_k$.

5 Main Results: Which TFMs Are MMIC, DSIC, or OCA-Proof?

This section develops general tools for reasoning about the incentive guarantees of different transaction fee mechanisms. We use six specific TFMs to illustrate our results:

1. A first-price auction (FPA), as described in Example 2.6.

2. A second-price-type auction (SPA), similar to Example 3.5. For concreteness, we assume that the intention is for a miner to order the outstanding transactions in non-increasing order of bid (per unit size) and include the largest feasible prefix of transactions. All included transactions pay the lowest accepted bid (per unit size), and all revenue is passed on to the miner.

3. A first-price auction in which a $\beta \in (0, 1]$ fraction of the transaction fees are burned (*β-burn FPA*)—that is, with $p_t(\mathbf{H}, B_k) = (1-\beta)b_t$ and $q_t(\mathbf{H}, B_k) = \beta b_t$ for an included transaction with bid b_t.

4. The 1559 mechanism, as described in Definition 4.1.

[27] Ties between subsets are intended to be broken consistently and independently of transactions' fee caps.

5. The *β-burn 1559 mechanism*, in which a $\beta \in [0,1)$ fraction of the base fee revenues are burned and the rest are passed on to a block's miner—that is, with $p_t(\mathbf{H}, B_k) = b_t - \beta r$ and $q_t(\mathbf{H}, B_k) = \beta r$ for an included transaction with bid b_t (where $r = \alpha(\mathbf{H})$). The intended allocation rule is analogous to that of the 1559 mechanism, with transactions chosen to maximize
$$\sum_{t \in M \,:\, b_t \geq r} x_t(\mathbf{H}, M) \cdot (b_t - \beta r - \mu) \cdot s_t.$$

6. The tipless mechanism, as described in Definition 4.2.

Table 1 summarizes the implications of our results for these six designs.

TFM	MMIC?	DSIC?	OCA-proof?
FPA	yes (Cor. 5.3)	no (obvious)	yes (Cor. 5.12)
SPA	no (Ex. 3.5)	almost[28]	almost (Rem. 5.13)
β-burn FPA	yes (Cor. 5.3)	no (obvious)	no (Cor. 5.17)
1559	yes (Cor. 5.3)	usually (Thm. 5.7)	yes (Cor. 5.14)
β-burn 1559	yes (Cor. 5.3)	usually (Rem. 5.9)	no (Cor. 5.18))
tipless	yes (Cor. 5.3)	yes (Thm. 5.5)	usually (Cor. 5.15+Rem. 5.16)

Table 1: Which of the six listed TFMs are MMIC, DSIC, or OCA-proof.

Thus, if we assess these TFMs solely according to these three types of incentive guarantees, FPAs dominate β-burn FPAs with $\beta > 0$; the 1559 mechanism dominates these and β-burn 1559 mechanisms with $\beta < 1$; and the 1559 mechanism and the tipless mechanism are incomparable.

5.1 MMIC and Non-MMIC TFMs

The MMIC condition (Definition 3.4) states that a revenue-maximizing miner should be incentivized to follow the intended allocation rule. Example 3.5 shows that not all interesting mechanisms are MMIC, and in particular that second-price-type auctions do not satisfy the condition. What goes wrong in Example 3.5 is that the payment collected from one transaction depends on the other included transactions. We call a payment rule *separable* if this is not the case.

Definition 5.1 (Separable Payment Rule) A payment rule $\mathbf{p}$ is *separable* if, for every on-chain history $\mathbf{H}$ and block B_k, the payment $p_t(\mathbf{H}, B_k)$ of an included transaction $t \in B_k$ is independent of the set $B_k - \{t\}$ of other included transactions and their bids.

[28] An SPA that uses the lowest included bid as a proxy for the highest excluded bid is not generally DSIC. However, it is approximately DSIC (with truthful bidding always nearly maximizing bidder utility) whenever these two values are close (as one would expect in a block with hundreds of transactions).

For a separable payment rule **p** and a fixed transaction t (with some bid b_t), we can write $p_t(\mathbf{H})$ for the payment that t's creator would pay should t be included in the block B_k that follows the history **H**. (This notation is well defined by separability.)

A separable payment rule **p** suggests a corresponding *revenue-maximizing* allocation rule **x**, in which a miner always chooses the most profitable subset of transactions. That is, given on-chain history **H** and a mempool M, the x_t's are assigned 0-1 values to maximize

$$\sum_{t \in M} x_t(\mathbf{H}, M) \cdot (p_t(\mathbf{H}) - \mu) \cdot s_t, \tag{8}$$

subject to feasibility.

Every TFM that uses a separable payment rule and the corresponding revenue-maximizing allocation rule is MMIC.

Theorem 5.2 (Separable Payments and Revenue Maximization Imply MMIC) *If* **p** *is a separable payment rule,* **x** *is the corresponding revenue-maximizing allocation rule, and* **q** *is an arbitrary burning rule, then the TFM* $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ *is MMIC.*

Proof: Fix an on-chain history **H**, a mempool M, and a marginal cost $\mu \geq 0$. By Definition 4.1, myopic miner utility (5) equals

$$u(F, B_k) := \underbrace{\sum_{t \in B_k \cap M} (p_t(\mathbf{H}) - \mu) \cdot s_t}_{\text{revenue less marginal costs}} - \underbrace{\sum_{t \in B_k \cap F} (\mu + q_t(\mathbf{H}, B_k)) \cdot s_t}_{\text{fake transaction costs}}, \tag{9}$$

where B_k denotes the transactions included by the miner and F the fake transactions that it creates. Included fake transactions can only increase the second term (as μ and **q** are nonnegative) while leaving the first unaffected (because **p** is separable), and so a myopic miner can be assumed to include only real transactions in B_k. In this case, myopic miner utility equals

$$\sum_{t \in B_k} (p_t(\mathbf{H}) - \mu) \cdot s_t,$$

which is identical to the quantity (8) maximized by the revenue-maximizing allocation rule. Thus, myopic miner utility is maximized by following the allocation rule and setting B_k equal to $\mathbf{x}(\mathbf{H}, M)$. ■

FPAs use the separable payment rule with $p_t(\mathbf{H}) = b_t$ and the corresponding revenue-maximizing rule (see (4)). β-burn FPAs use the separable rule $p_t(\mathbf{H}) = (1-\beta)b_t$ and the same revenue-maximizing allocation rule. The 1559, β-burn 1559, and tipless mechanisms use the separable payment rules given by $p_t(\mathbf{H}) = b_t - r$, $p_t(\mathbf{H}) = b_t - \beta r$, and $p_t(\mathbf{H}) = \delta$, respectively, where r denotes the 1559 mechanism's current base fee (which, crucially, depends only on **H**) and δ is the hard-coded tip in the tipless mechanism. By definition, all three mechanisms use the corresponding revenue-maximizing allocation rules. Applying Theorem 5.2:

Corollary 5.3 (Five MMIC TFMs) *FPAs, β-burn FPAs, the 1559 mechanism, the β-burn 1559 mechanism, and the tipless mechanism are all MMIC.*

Remark 5.4 (Non-Separable MMIC TFMs) There exist MMIC TFMs that do not employ a separable payment rule. For example, define $\mathbf{q}$ as the all-zero function, $p_t(\mathbf{H}, B_k) = b_t$ if t is the highest-bidding transaction included in B_k (with ties broken arbitrarily but consistently), and $p_t(\mathbf{H}, B_k) = 0$ otherwise. Define $\mathbf{x}$ as the rule that includes only the highest-bidding transaction in the mempool (or, if every bid is less than μ, includes nothing). This TFM is MMIC even though $\mathbf{p}$ is not separable. The monopolistic price mechanism proposed by Lavi et al. [32] and discussed in Section 1 is another example. Characterizing the MMIC TFMs is an interesting open research question.

5.2 DSIC and Non-DSIC TFMs

The DSIC condition (Section 3.1) states that every transaction creator should always have a dominant bidding strategy—a bid that maximizes their utility (3), no matter what the bids of others. The optimal bid in an FPA or β-burn FPA depends on others' bids, so these TFMs are not DSIC.

The tipless mechanism is an example of a DSIC TFM.

Theorem 5.5 *The tipless mechanism is DSIC.*

Proof: Fix an on-chain history $\mathbf{H}$ and corresponding base fee $r = \alpha(\mathbf{H})$. The claim is that the bidding strategy $\sigma(v_t) = \min\{r + \delta, v_t\}$ is a dominant strategy for every bidder, where δ denotes the value of the hard-coded tip in the tipless mechanism.[29]

Fix a set T of transactions with valuations $\mathbf{v}$ and a transaction $t \in T$. Suppose t's creator bids $b_t = \sigma(v_t) = \min\{r + \delta, v_t\}$. If t is a low-value transaction (with $v_t < r + \delta$), every alternative bid $\hat{b}_t$ either has no effect on t's utility or leads to t's inclusion in the block; the latter occurs only when $\hat{b}_t \geq r + \delta$, in which case the creator's utility drops from 0 to $(v_t - \hat{b}_t) \cdot s_t < 0$. For a high-value transaction (with $v_t \geq r + \delta$), every alternative bid $\hat{b}_t$ either has no effect on the creator's utility or, if the alternative bid triggers t's exclusion, drops its utility from a nonnegative number $(v_t - r - \delta) \cdot s_t \geq 0$ to 0.[30] We conclude that the bid $\sigma(v_t) = \min\{r + \delta, v_t\}$ is always utility-maximizing for t's creator. ■

The 1559 mechanism is not DSIC in general, as for the special case of a zero base fee it is equivalent to an FPA. However, given that the base fee is automatically adjusted over time in response to excess demand, one might expect that, in a typical block, the base fee is

[29] In the mechanism's implementation (Section 4), this is precisely the bid induced by a truthfully reported fee cap (with $c_t = v_t$) and the current base fee r.

[30] Recall that the tipless mechanism's allocation rule includes a largest-possible subset of the eligible transactions (i.e., transactions t with $b_t \geq r + \delta$), breaking ties between subsets in a consistent and bid-independent way. (Unless $\delta < \mu$, in which case the mechanism never includes any transactions and is trivially DSIC.)

sufficiently high to exclude all but a reasonable number of transactions in the mempool. Can we at least argue that, if the base fee "does its job," then the 1559 mechanism is DSIC?[31]

Definition 5.6 (Excessively Low Base Fee) Let μ denote the marginal cost. In the 1559 mechanism, a base fee r is *excessively low* for a set T of transactions with valuations $\mathbf{v}$ if the demand at price $r+\mu$ exceeds the maximum block size C_{max}:

$$\underbrace{\sum_{t \in M \,:\, v_t \geq r+\mu} s_t}_{\text{demand at price } r+\mu} > C_{max}. \tag{10}$$

Are excessively low base fees the only obstruction to the DSIC property? Not quite. The issue is that if, for whatever reason, users choose to overbid (with $b_t > v_t$), a base fee may act as if it is excessively low (with respect to the reported bids) even though it is not (with respect to the true valuations). The next result proves that these are the only two obstructions to achieving DSIC—without them, the 1559 mechanism effectively acts as a posted price mechanism with a price (per unit size) equal to the base fee r plus the miner marginal cost μ.

Theorem 5.7 (The 1559 Mechanism Is Usually DSIC) *Fix an on-chain history* $\mathbf{H}$ *and corresponding base fee* $r = \alpha(\mathbf{H})$*, a marginal cost* μ*, and a set* T *of transactions with valuations* $\mathbf{v}$*. If* r *is not excessively low for* T *and transaction creators cannot overbid, the bidding strategy* $b_t = \sigma(v_t) = \min\{r+\mu, v_t\}$ *is a dominant strategy for every bidder.*[32]

Proof: Fix a transaction $t \in T$, with valuation v_t. Suppose first that $v_t < r+\mu$. The objective (6) of the 1559 allocation rule prescribes including only transactions $s \in T$ with $b_s \geq r+\mu$. If t's creator bids $b_t = \min\{r+\mu, v_t\} = v_t$, the transaction will be excluded from the block and the resulting utility will be 0. Every alternative bid $\hat{b}_t$ either leads to the same outcome or, if it results in t's inclusion in the block, leads to negative utility (at most $v_t - (r+\mu) < 0$).

Now suppose that $v_t \geq r+\mu$. Because transaction creators cannot overbid, the transactions $w \in T$ with $b_w \geq r+\mu$ are a subset of the transactions $w \in T$ with $v_w \geq r+\mu$. Thus, because r is not excessively low for T, there is room for all of these transactions (no matter what b_t is):

$$\underbrace{\sum_{w \in T \,:\, b_w \geq r+\mu} s_w}_{\text{total size of included txs}} \leq \underbrace{\sum_{w \in T \,:\, v_w \geq r+\mu} s_w}_{\text{demand at price } r+\mu} \leq C_{max}. \tag{11}$$

[31] A natural conjecture is that, for an appropriately tuned base fee update rule α, excessively low base fees should arise only in short transitory periods while waiting for the base fee to catch up to a large and sudden demand spike. Preliminary investigations by Monnot [37] with synthetic data provide some initial support for this conjecture.

[32] In the mechanism's implementation (Section 4), this is precisely the bid induced by a truthfully reported fee cap (with $c_t = v_t$), a tip of μ, and a current base fee of r.

If t's creator bids $b_t = \min\{r + \mu, v_t\} = r + \mu$, the transaction will be included in the block and the resulting utility will be $v_t - (r + \mu) \geq 0$. Every alternative bid either leads to t's exclusion (resulting in utility 0) or to t's inclusion at a price higher than $r + \mu$. We conclude that there is no alternative bid for t that increases its creator's utility, and hence $\sigma(v_t) = \min\{r + \mu, v_t\}$ is a dominant bidding strategy. ■

Remark 5.8 (Alternative Interpretation of Theorem 5.7) The proof of Theorem 5.7 shows that, whenever the base fee is not excessively low and all other transaction creators do not overbid (e.g., on account of overbidding being a dominated strategy), a best response of a transaction creator with valuation v_t is to bid $\sigma(v_t) = \min\{r + \mu, v_t\}$. Thus, when the base fee is not excessively low, the outcome in which all transaction creators bid according to the strategy σ constitutes an ex post Nash equilibrium.

Remark 5.9 (The β-Burn 1559 Mechanism) The β-burn 1559 and 1559 mechanisms make the same allocation decisions and charge the same total price (i.e., miner payment plus burn) for included transactions, and differ only in how the payments by users are directed. Thus the two mechanisms are identical from the user perspective, and Theorem 5.7 carries over immediately to all β-burn 1559 mechanisms.

5.3 OCA-Proof and Non-OCA-Proof TFMs

The OCA-proofness condition (Definition 3.7) requires the existence of a canonical and individually rational on-chain outcome that cannot be Pareto improved by any off-chain agreement (specifying the on-chain allocation and bids, and the off-chain transfers between users and the block's miner).

Because OCAs can specify arbitrary transfers, we can characterize OCA-proofness in terms of a surplus-maximization property. The next definition is the sum of the utility functions of the miner and all the creators of pending transactions.

Definition 5.10 (Joint Utility) For an on-chain history $\mathbf{H}$ and mempool M, the *joint utility* of the miner and the creators of transactions in M for a block B_k is

$$\sum_{t \in B_k} (v_t - q_t(\mathbf{H}, B_k) - \mu) \cdot s_t. \tag{12}$$

From the perspective of a coalition of users and a miner, on-chain and off-chain payments from the users to the miner (the p_t's specified by the TFM and the τ_t's specified by the OCA) remain within the coalition and cancel out; burned money (the q_t's) is transferred outside the coalition and is therefore a loss.

Proposition 5.11 (OCA-Proof $\Leftrightarrow$ Joint Utility-Maximization) *A TFM* $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ *is OCA-proof if and only if, for every on-chain history* $\mathbf{H}$*, there exists an individually rational bidding strategy* $\sigma_{\mathbf{H}}$ *such that, for every possible set* T *of outstanding transactions and valuations* $\mathbf{v}$*, the outcome* $B_k = \mathbf{x}(\mathbf{H}, T(\sigma_{\mathbf{H}}(\mathbf{v})))$ *maximizes the joint utility* (12) *over every possible on-chain outcome* $\mathbf{x}(\mathbf{H}, T(\mathbf{b}))$.

Proof: For the "only if" direction, suppose there exists a history $\mathbf{H}$ such that, for every individually rational bidding strategy $\sigma_{\mathbf{H}}$, there is a set T of transactions with valuations $\mathbf{v}$ and a set $\mathbf{b}$ of on-chain bids such that the joint utility of the outcome $\mathbf{x}(\mathbf{H}, T(\mathbf{b}))$ is strictly larger than that of the outcome $\mathbf{x}(\mathbf{H}, T(\sigma_{\mathbf{H}}(\mathbf{v})))$. Then, there is an OCA that uses on-chain bids $\mathbf{b}$ and suitable transfers to share the additional joint utility (relative to the outcome $\mathbf{x}(\mathbf{H}, T(\sigma_{\mathbf{H}}(\mathbf{v}))))$ such that all transaction creators and the miner are strictly better off.

Conversely, suppose that for every history $\mathbf{H}$, there exists an individually rational bidding strategy $\sigma_{\mathbf{H}}$ such that, for every set T of transactions and valuation vector $\mathbf{v}$, the bid profile $\sigma_{\mathbf{H}}(\mathbf{v})$ maximizes the joint utility over all possible on-chain bids $\mathbf{b}$. Then, for every OCA, its joint utility is at most that of $\mathbf{x}(\mathbf{H}, T(\sigma_{\mathbf{H}}(\mathbf{v})))$; because the joint utility is the sum of the miner and users' utility functions, some participant's utility under the OCA is at most that in $\mathbf{x}(\mathbf{H}, T(\sigma_{\mathbf{H}}(\mathbf{v})))$. Thus, no OCA Pareto improves over the outcome with on-chain bids $\sigma_{\mathbf{H}}(\mathbf{v})$ and no off-chain transfers. ■

Proposition 5.11 reduces the task of verifying OCA-proofness to checking whether or not there is a joint utility-maximizing and individually rational bidding strategy. We proceed to check each of our six running examples.

Corollary 5.12 *FPAs are OCA-proof.*

Proof: Fix an (irrelevant) history $\mathbf{H}$. Let $\gamma \in (0, 1]$ be arbitrary, and define a bidding strategy σ by

$$\sigma(v_t) = \min\{v_t, \mu + \gamma\,(v_t - \mu)\}.$$

For a first-price auction, this bidding strategy is individually rational. Consider a set T of transactions with valuations $\mathbf{v}$. Because the burning rule $\mathbf{q}$ is the all-zero function, the objective (4) maximized by the allocation rule $\mathbf{x}^f$ with bids $\sigma(\mathbf{v})$ is identical (modulo the scaling factor γ) to the joint utility (12). Thus, the joint utility of the on-chain outcome with bids $\sigma(\mathbf{v})$ cannot be improved upon by any OCA. ■

Remark 5.13 (SPAs Are Almost OCA-Proof) In an SPA, the burning rule $\mathbf{q}$ is the all-zero function and the joint utility (12) reduces to the social welfare $\sum_{t \in B_k} (v_t - \mu) \cdot s_t$. Had we defined an SPA using the welfare-maximizing allocation rule, it would be an OCA-proof transaction fee mechanism (by Proposition 5.11, using the identity bidding strategy $\sigma(v_t) = v_t$). For our definition of SPAs based on a greedy heuristic allocation rule, the outcome may have slightly less than the maximum-possible social welfare, and so (by Proposition 5.11) there may be an opportunity for a (small) Pareto improvement.

The proof that FPAs are OCA-proof (Corollary 5.12) can be extended to the 1559 mechanism.

Corollary 5.14 *The 1559 mechanism is OCA-proof.*

Proof: Fix a history $\mathbf{H}$ and corresponding base fee $r = \alpha(\mathbf{H})$. Let $\gamma \in (0, 1]$ be arbitrary, and define a bidding strategy σ by

$$\sigma(v_t) = \min\{v_t, \mu + r + \gamma\,(v_t - \mu - r)\}.$$

This bidding strategy is individually rational. Consider a set T of transactions with valuations $\mathbf{v}$. The objective (6) maximized by the allocation rule $\mathbf{x}^*$ with bids $\sigma(\mathbf{v})$ is identical (modulo the scaling factor γ) to the joint utility (12). Thus, the joint utility of the on-chain outcome with bids $\sigma(\mathbf{v})$ cannot be improved upon by any OCA, and Proposition 5.11 then implies that the mechanism is OCA-proof. ■

The tipless mechanism fails OCA-proofness in the same regime in which the 1559 mechanism loses its DSIC property, the regime of an excessively low base fee (Definition 5.6). In effect, the tipless mechanism retains DSIC in this regime by disallowing bidders to differentiate themselves through high bids, and fails OCA-proofness for the same reason.

Corollary 5.15 (The Tipless Mechanism Is Usually OCA-Proof) *Fix an on-chain history* $\mathbf{H}$ *and corresponding base fee* $r = \alpha(\mathbf{H})$*, and a set* T *of transactions with valuations* $\mathbf{v}$ *such that* r *is not excessively low. The tipless mechanism, with hard-coded tip* δ *equal to the marginal cost* μ*, is OCA-proof.*

Proof: Define an (individually rational) bidding strategy σ by $\sigma(v_t) = \min\{r + \delta, v_t\} = \min\{r + \mu, v_t\}$. The joint utility (12) of the miner and users for the outcome B_k is

$$\sum_{t \in B_k} (v_t - r - \mu) \cdot s_t. \tag{13}$$

Because r is not excessively low for T, the total size of the transactions t with $v_t \geq r+\mu = r+\delta$ is at most the maximum block size C_{max}. The joint utility (13) is therefore maximized by including these (and only these) transactions, which is precisely the outcome under the bids $\sigma(\mathbf{v})$. Proposition 5.11 then implies that, under the assumption of a base fee that is not excessively low, the tipless mechanism is OCA-proof. ■

Remark 5.16 (The Tipless Mechanism Is Not Generally OCA-Proof) The tipless mechanism is not generally OCA-proof when the base fee r is excessively low (even with $\delta = \mu$). For consider an arbitrary individually rational bidding strategy σ. If $\sigma(v) < r + \delta$ for some $v > r + \delta$, a collection of transaction creators all with valuation v would be better off in an OCA with a miner than bidding $\sigma(v)$ on-chain (which would lead to the automatic exclusion of all the transactions). On the other hand, if $\sigma(v) \geq r + \delta$ for all $v > r + \delta$, consider a collection of transactions T with total size $\sum_{t \in T} s_t$ bigger than the maximum block size C_{max} and with creators that have distinct valuations, all bigger than $r + \delta$. The intended allocation rule then instructs the miner to include a subset of transactions with the maximum-possible total size (subject to the block capacity C_{max}). This will not generally be a subset $S \subseteq T$ of transactions that maximizes the joint utility $\sum_{t \in S}(v_t - r - \mu) \cdot s_t$

subject to the block capacity; by Proposition 5.11, such an outcome can be Pareto improved through an OCA.

The tipless mechanism is also not generally OCA-proof when $\delta > \mu$: a collection of transaction creators all with the same valuation $v \in (r + \mu, r + \delta)$ would be better off in an OCA with a miner than bidding on-chain according to an individually rational bidding strategy (which would lead to the automatic exclusion of all the transactions). If $\delta - \mu$ is small, however, the improvement in joint surplus possible through an OCA will be negligible.

Two of the biggest differences between an FPA and the 1559 mechanism are the switch to a posted-price-type mechanism and the burning of transaction fees. The OCA-proofness considerations in our final two corollaries explain why it's important to make both changes simultaneously, rather than either one in isolation.

Corollary 5.17 *For every $\beta \in (0, 1]$, a β-burn FPA is not OCA-proof.*

Proof: Assume that $\mu = 0$; the general case is similar. Fix an (irrelevant) history $\mathbf{H}$, and consider an arbitrary bidding strategy $\sigma_{\mathbf{H}}$. If $\sigma_{\mathbf{H}}$ is the all-zero function, the allocation rule instructs the miner to include an arbitrary feasible subset of transactions (breaking ties arbitrarily but consistently), which will not generally be a joint utility-maximizing feasible subset. So suppose that $\sigma_{\mathbf{H}}(v_t) > 0$ for some $v_t > 0$. Then, if the mempool is a single transaction with value v_t, the joint utility of the on-chain outcome with bid $\sigma_{\mathbf{H}}(v_t)$ is $v_t - \beta \cdot \sigma_{\mathbf{H}}(v_t)$. Because $\beta > 0$, the joint utility would be higher with any smaller on-chain bid $b_t \in (0, \sigma_{\mathbf{H}}(v_t))$. We conclude that no bidding strategy (individually rational or otherwise) is guaranteed to maximize the joint utility, and hence (by Proposition 5.11) this TFM is not OCA-proof. ■

Corollary 5.18 *For every $\beta \in [0, 1)$, the β-burn 1559 mechanism is not OCA-proof.*

Proof: Fix $\beta \in [0, 1)$ and a history $\mathbf{H}$ that leads to a positive base fee $r = \alpha(\mathbf{H}) > 0$. Consider an arbitrary individually rational bidding strategy $\sigma_{\mathbf{H}}$. Because transaction creators are charged their bids in the β-burn 1559 mechanism, individual rationality implies that there is no overbidding: $\sigma_{\mathbf{H}}(v_t) \leq v_t$ for every $v_t \geq 0$. Now consider a mempool with one transaction t with a valuation v_t that is strictly between $\beta r + \mu$ and $r + \mu$, where μ denotes the marginal cost; v_t exists because $\beta < 1$ and $r > 0$. Because $v_t < r + \mu$, the suggested bid $\sigma_{\mathbf{H}}(v_t)$ would lead to t's exclusion and a joint utility of 0. The bid $b_t = r + \mu$ would lead to a joint utility of $v_t - \beta r - \mu$ which, because $v_t > \beta r + \mu$, is strictly positive. Proposition 5.11 now implies that the mechanism is not OCA-proof. ■

6 Discussion

The 1559 mechanism vs. the tipless mechanism, and beyond. The 1559 and tipless mechanisms are the two clear winners in our study of MMIC, DSIC, and OCA-proof transaction fee mechanisms—both provide two of these three incentive-compatibility guarantees in all circumstances, and all three in what is plausibly the common case of a base fee that is reasonably well tuned to the current demand. In a block with an excessively low base fee (presumably due to rapidly increasing demand), one might expect bidding wars in the 1559 mechanism (on account of the failure of DSIC) and off-chain coordination in the tipless mechanism (due to the breakdown of OCA-proofness). We conjecture that there is no transaction fee mechanism that always satisfies all three properties.[33] More generally, it would be interesting to characterize the mechanisms that satisfy various subsets and relaxations of these three properties.

Perhaps the strongest argument in favor of the tipless mechanism over the 1559 mechanism is its simplicity. On the user side, there are several simplifications. The creator of a transaction t must specify only one parameter (a fee cap c_t) rather than two (a fee cap c_t and a tip δ_t). The "obvious optimal bid" in the tipless mechanism (setting $c_t = v_t$) is optimal for every block and no matter what the bids of the competing transactions. The "obvious optimal bid" in the 1559 mechanism (setting $c_t = v_t$ and $\delta_t = \mu$) is optimal only in blocks with neither an excessively low base fee nor overbidding. On the miner side, assuming that $\delta \geq \mu$, the revenue-maximizing strategy simplifies to maximizing the block size while using only eligible transactions (i.e., with bid at least $r + \delta$). On the negative side, the hard-coded tip δ in the tipless mechanism would likely need to be adjusted over time via protocol upgrades. Also, the tipless mechanism's apparent DSIC advantage over the 1559 mechanism in blocks with excessively low base fees breaks down in the presence of cartels of colluding users or even a single user who creates multiple transactions. The reason is that, in a block with an excessively low base fee, such a cartel or user could coordinate bids across transactions to manipulate the (arbitrary but consistent) tie-breaking rule of the mechanism. When the base fee is not excessively low, the 1559 and tipless mechanisms are effectively unlimited-supply posted-price mechanisms; as such, the bidding strategies in Theorems 5.5 and 5.7 remain optimal for users that control multiple transactions (assuming an additive valuation over them) and for cartels of colluding users.

Alternatives to burning. Corollary 5.18 shows that, for a block's base fee to be economically meaningful, no revenue from it can be passed on to the block's miner. The simplest solution is to burn all the base fee revenues. One appealing alternative implementation, with the same incentive-compatibility properties, is to instead split the base fee revenues of a block equally among the miners of the next ℓ blocks. (Here ℓ is a tunable parameter; the 1559 mechanism can be thought of as the special case in which $\ell = 0$.) Thus, a miner of a block receives a $1/\ell$ fraction of the sum of the base fee revenues from the previous ℓ blocks, along with all of the tips from the current block. While burning the base fee revenue favors holders of the native currency (through deflation), this "pay-it-forward" implementation favors miners (through more transaction fee revenue). A second trade-off between the two implementations concerns whether variability in demand (and hence transaction fee revenue) translates to variability in blockchain security or in the issuance of new currency. With money-burning, every block potentially changes the money supply in two ways: minting new

[33] As discussed in Section 1, Chung and Shi [17] prove such an impossibility result under a collision-resistance requirement different than OCA-proofness. The relative weakness of the OCA-proof condition appears to be the primary challenge in proving this conjecture.

coins for inflationary rewards (like a block reward), and burning the coins used to pay the base fee. Thus the blockchain's inflation rate would be variable and unpredictable from block to block, but miner revenue (which effectively pays for the blockchain's security [3, 8]) would stay relatively constant (modulo fluctuations in the market price of the native currency). With the pay-it-forward implementation, the inflation rate would be essentially deterministic but the miner rewards (and, hence, blockchain security) would be unpredictable (though never less than that with money-burning).

Multi-block time scales. This paper focuses on incentive issues in transaction fee mechanisms at the time scale of a single block. Many candidate deviations by miners and users manifest already at this time scale. The 1559 mechanism and its variants entangle the transaction fee mechanisms for different blocks through a history-dependent base fee. Dependencies between blocks open up the possibility of miner deviations that unfold over longer time scales.

For example, publishing a smaller-than-target block decreases the base fee for the next block, potentially increasing the revenue of that block's miner. Every miner would happily free ride on previous miners who have sacrificed some eligible transactions to keep block sizes and hence the base fee down, but would rather not make such a sacrifice themselves. Long-term manipulation by a cartel of miners thus resembles the challenge of sustaining collusion in a repeated multi-player Prisoner's Dilemma game: If all players cooperate (e.g., keep block sizes small to keep the base fee low in the 1559 mechanism or the bids high in an FPA) they are all better off, but each player has a myopic incentive to unilaterally deviate from this strategy (e.g., when mining a block, pack it full to maximize the immediate tip revenue, thereby decreasing the revenue earned by miners of future blocks).

Persistent and harmful miner collusion has not yet been definitively observed in a major blockchain such as Bitcoin or Ethereum. None of the primary transaction fee mechanisms discussed in this paper (FPAs, the 1559 mechanism, and the tipless mechanism) are obviously more vulnerable than the others to such long-term collusion. It would be interesting to develop a more nuanced understanding of this issue, especially in a regime in which a single miner or validator might control a significant fraction (30%, say) of the overall hashrate or stake and therefore might plausibly benefit from non-myopic strategies.[34]

References

[1] G. Aggarwal and J. D. Hartline. Knapsack auctions. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1083–1092, 2006.

[2] M. Akbarpour and S. Li. Credible auctions: A trilemma. *Econometrica*, 88(2):425–467, 2020. URL: `http://web.stanford.edu/~mohamwad/CredibleMechanisms.pdf`.

[34] Non-myopic validator strategies might be particularly effective in proof-of-stake blockchain protocols that, like the current version of Ethereum, grant validators some degree of advance knowledge about which blocks they will be chosen to produce.

[3] R. Auer. Beyond the doomsday economics of “proof-of-work” in cryptocurrencies. BIS working paper #765. URL: https://www.bis.org/publ/work765.pdf, January 2019.

[4] S. Azouvi, G. Goren, L. Heimbach, and A. Hicks. Base fee manipulation in Ethereum’s EIP-1559 transaction fee mechanism. In *Proceedings of the 37th International Symposium on Distributed Computing (DSIC)*, 2023. Article 6.

[5] M. Bahrani, P. Garimidi, and T. Roughgarden. Transaction fee mechanism design with active block producers. arXiv:2307.01686. URL: https://arxiv.org/pdf/2307.01686.pdf, July 2023.

[6] S. Basu, D. Easley, M. O’Hara, and E. G. Sirer. Stablefees: A predictable fee market for cryptocurrencies. *Management Science*, 69(11):6508–6524, 2023.

[7] J. Bier. *The Blocksize War: The battle over who controls Bitcoin’s protocol rules.* Amazon KDP, 2021.

[8] E. Budish. Trust at scale: The economic limits of cryptocurrencies and blockchains. Working paper. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4148014, 2023.

[9] V. Buterin. Ethereum: A next-generation smart contract and decentralized application platform. Unpublished white paper. URL: https://ethereum.org/en/whitepaper/, November 2013.

[10] V. Buterin. Blockchain resource pricing. URL: https://ethresear.ch/uploads/default/original/2X/1/197884012ada193318b67c4b777441e4a
August 2018.

[11] V. Buterin. First and second-price auctions and improved transaction-fee markets. URL: https://ethresear.ch/t/first-and-second-price-auctions-and-improved-transaction-fee-
July 2018.

[12] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. EIP-1559 specification. URL: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md, 2019.

[13] A. Canidio. Auctions with tokens: Monetary policy as a mechanism design choice. arXiv:2301.13794. URL: https://arxiv.org/pdf/2301.13794.pdf, August 2023.

[14] M. Carlsten, H. A. Kalodner, S. M. Weinberg, and A. Narayanan. On the instability of Bitcoin without the block reward. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 154–167, 2016. URL: https://www.cs.princeton.edu/~arvindn/publications/mining_CCS.pdf.

[15] X. Chen, D. Simchi-Levi, Z. Zhao, and Y. Zhou. Bayesian mechanism design for blockchain transaction fee allocation. arXiv:2209.13099. URL: `https://arxiv.org/pdf/2209.13099.pdf`, September 2022.

[16] T. Chitra, M. V. X. Ferreira, and K. Kulkarni. Credible, optimal auctions via blockchains. arXiv:2301.12532. URL: `https://arxiv.org/pdf/2301.12532.pdf`, January 2023.

[17] H. Chung and E. Shi. Foundations of transaction fee mechanism design. In *Proceedings of the 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3856–3899, 2023.

[18] D. Crapis, C. C. Moallemi, and S. Wang. Optimal dynamic fees for blockchain resources. arXiv:2309.12735. URL: `https://arxiv.org/pdf/2309.12735.pdf`, September 2023.

[19] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels. Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *Proceedings of the 41st IEEE Symposium on Security and Privacy*, pages 910–927, 2020. URL: `https://arxiv.org/pdf/1904.05234.pdf`.

[20] C. Daskalakis, M. Fishelson, B. Lucier, V. Syrgkanis, and S. Velusamy. Simple, credible, and approximately-optimal auctions. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, page 713, 2020.

[21] C. Decker and R. Wattenhofer. Information propagation in the Bitcoin network. In *13th IEEE International Conference on Peer-to-Peer Computing*, 2013. URL: `https://tik-db.ee.ethz.ch/file/49318d3f56c1d525aabf7fda78b23fc0/P2P2013_041.pdf`.

[22] Easley, M. O'Hara, and S. Basu. From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics*, 134(1):91–109, October 2019. URL: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3055380`.

[23] M. V. X. Ferreira, D. J. Moroz, D. C. Parkes, and M. Stern. Dynamic posted-price mechanisms for the blockchain transaction-fee market. In *Proceedings of the 3rd ACM Advances in Financial Technologies*, pages 86–99, 2021.

[24] M. V. X. Ferreira and D. C. Parkes. Credible decentralized exchange design via verifiable sequencing rules. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 723–723, 2023.

[25] Y. Gafni and A. Yaish. Greedy transaction fee mechanisms for (non-)myopic miners. arXiv:2210.07793. URL: `https://arxiv.org/pdf/2210.07793.pdf`, October 2022.

[26] Hasu, J. Prestwich, and B. Curtis. A model for Bitcoin's security and the declining block subsidy. Working paper. URL: `https://uncommoncore.co/wp-content/uploads/2019/10/A-model-for-Bitcoins-security-and`
October 2019.

[27] J. L. Hougaard and M. Pourpouneh. Farsighted miners under transaction fee mechanism EIP1559. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9, 2023.

[28] N. Houy. The economics of Bitcoin transaction fees. Working paper #1407, Groupe d'Analyse et de Théorie Economique Lyon St-Étienne (GATE Lyon St-Étienne), Université de Lyon. URL: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2400519`, February 2014.

[29] G. Huberman, J. Leshno, and C. C. Moallemi. Monopoly without a monopolist: An economic analysis of the Bitcoin payment system. *Review of Economic Studies*, 88(6):3011–3040, November 2021.

[30] A. Kiayias, E. Koutsoupias, P. Lazos, and G. Panagiotakos. Tiered mechanisms for blockchain transaction fees. arXiv:2304.06014. URL: `https://arxiv.org/pdf/2304.06014.pdf`, April 2023.

[31] A. Kiayias, P. Lazos, and J. C. Schlegel. Would Friedman burn your tokens? In *Proceedings of the 27th Conference on Financial Cryptography and Data Security (FC)*, 2024. To appear. URL: `https://arxiv.org/pdf/2306.17025.pdf`.

[32] R. Lavi, O. Sattath, and A. Zohar. Redesigning Bitcoin's fee market. *ACM Transactions on Economics and Computation*, 10(1), 2022. Article 5.

[33] S. Leonardos, B. Monnot, D. Reijsbergen, S. Skoulakis, and G. Piliouras. Dynamical analysis of the EIP-1559 Ethereum fee market. In *Proceedings of the 3rd ACM Advances in Financial Technologies*, pages 114–126, 2021.

[34] S. Leonardos, D. Reijsbergen, B. Monnot, and G. Piliouras. Optimality despite chaos in fee markets. In *Proceedings of the 27th Conference on Financial Cryptography and Data Security (FC)*, pages 346–362, 2023.

[35] X. Liu, Y. Liu, H. Li, J. Wang, J. Zhu, and H. Song. Multi-side incentive compatible transaction fee mechanism. *Computers and Electrical Engineering*, 113:109050, 2024.

[36] Y. Liu, Y. Lu, K. Nayak, F. Zhang, L. Zhang, and Y. Zhao. Empirical analysis of EIP-1559: Transaction fees, waiting times, and consensus security. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2099–2113, 2022.

[37] B. Monnot. EIP 1559: A transaction fee market proposal. URL: `https://ethereum.github.io/abm1559/notebooks/eip1559.html`, April 2020.

[38] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[39] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Unpublished white paper. URL: https://bitcoin.org/bitcoin.pdf, 2008.

[40] A. Ndiaye. Blockchain price vs. quantity controls. Working paper. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4538155, July 2023.

[41] N. Nisan. Serial monopoly on blockchains. arXiv:2311.12731. URL: https://arxiv.org/pdf/2311.12731.pdf, November 2023.

[42] D. Reijsbergen, S. Sridhar, B. Monnot, S. Leonardos, S. Skoulakis, and G. Piliouras. Transaction fees on a honeymoon: Ethereum's EIP-1559 one month later. In *Proceedings of the IEEE International Conference on Blockchain*, pages 196–204, 2021.

[43] P. R. Rizun. A transaction fee market exists without a block size limit. Block size limit debate working paper. URL: https://www.bitcoinunlimited.info/resources/feemarket.pdf, August 2015.

[44] T. Roughgarden. Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559. arXiv:2012.00854. URL: https://arxiv.org/pdf/2012.00854.pdf, December 2020.

[45] E. Shi, H. Chung, and K. Wu. What can cryptography do for decentralized mechanism design? arXiv:2209.14462. URL: https://arxiv.org/pdf/2209.14462.pdf, September 2022.

[46] W. Tang and D. D. Yao. Transaction fee mechanism for proof-of-stake protocol. arXiv:2308.13881. URL: https://arxiv.org/pdf/2308.13881.pdf, August 2023.

[47] K. Wu, E. Shi, and H. Chung. Maximizing miner revenue in transaction fee mechanism design. In *Proceedings of the 15th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2024. To appear. URL: https://eprint.iacr.org/2023/283.pdf.

[48] A. C.-C. Yao. An incentive analysis of some Bitcoin fee designs. arXiv:1811.02351. URL: https://arxiv.org/pdf/1811.02351.pdf, November 2018.

[49] L. Zhang and F. Zhang. Understand waiting time in transaction fee mechanism: An interdisciplinary perspective. arXiv:2305.02552. URL: https://arxiv.org/pdf/2305.02552.pdf, May 2023.