Short Paper: Unpredictable Transaction Arrangement for MEV Mitigation in Ethereum

Jan Droll
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
0009-0003-5003-3898

Oliver Stengele
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
0000-0002-0574-0628

Hannes Hartenstein
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
0000-0003-3441-3180

***Abstract*—We present a simple approach to mitigate transaction-ordering dependent Maximum-Extractable-Value (MEV) opportunities in Ethereum. The approach is built on top of Proposer-Builder-Separation, restricts the freedom to order transactions within a block arbitrarily, and relies primarily on a non-malleable signature scheme and separation of responsibilities. In addition, we generalize the proposed approach to show that the property of unpredictable transaction arrangement can be transferred also to other blockchain systems. We discuss the gain in MEV mitigation, important limitations, and open issues.**

***Index Terms*—Unpredictable Transaction Arrangement, Maximum Extractable Value, Proposer Builder Separation, Transaction ordering Dependence, Ethereum**

I. Introduction

Blockchain systems motivate active participation through financial incentives. In Ethereum [2] for example, proposers earn a block reward defined by the protocol and tips specified by senders of transactions included in the proposed block. Due to the presumed rational behavior of participants, proposers maximize their revenue during block creation. One opportunity for a proposer to maximize their revenue is to include transactions prioritized according to their specified tips. Further opportunities exist and are covered by the term Maximum-Extractable-Value (MEV) [5]. In this work, we focus on *transaction-ordering dependent (TOD)* opportunities [8]. Such opportunities are present especially in Decentralized Finance (DeFi) applications and exploitable due to the *degrees of freedom* during block creation. We present an approach on top of enshrined Proposer-Builder-Separation (ePBS) that establishes an arrangement for transactions within a block to be proposed. While all possible arrangements are determinable, the approach defines a single arrangement to be valid and in this way restrict the degrees of freedom. This valid arrangement remains unpredictable until the proposer decided which transactions are included in the next block. Key elements of the approach are a non-malleable signature scheme and separation of responsibilities. After presenting background and related work, we outline the proposed approach, called One-To-Three. We generalize the idea to a concept of unpredictable transaction arrangement and discuss the resulting mitigation, limitations, and open issues.

II. Background and Related Work

We assume the reader is familiar with Ethereum [2] as a decentralized system that serves as platform for decentralized applications, and, in particular, with the underlying Proof-of-Stake (PoS) consensus. In the following, we focus on the notions of MEV and Proposer-Builder-Separation (PBS), as well as on MEV countermeasures and mitigation approaches.

Various definitions for the notion of MEV exist, see, e.g. [5], [7]. As a general term, MEV refers to all potential value that can be gained through the degrees of freedom during block creation and through the public nature of blockchain data. In particular, consensus rules leave open the following degrees of freedom: *i)* which transactions to include in a block, *ii)* the arrangement of transactions within a block, and *iii)* whether transactions need to be published beforehand. The public nature of blockchain data cover the fact that transactions within the mempool, blocks, and the overall state of Ethereum are publicly accessible and, therefore, subject to analysis.

MEV opportunities can be categorized as protocol-defined, smart contracts-enabled, or censoring-motivated. Protocol-defined MEV opportunities are encoded in the consensus rules of Ethereum and are, for example, the block reward, tips of included transactions, and attestation rewards. Smart Contract-enabled (SC-enabled) MEV opportunities arise from transactions with smart contracts (SCs) or from SCs themselves. In particular, DeFi applications provide SC-enabled MEV opportunities that utilize slippage, arbitrage, or liquidations. Other SC-enabled MEV opportunities are vulnerabilities in SCs or challenges and games offered via SCs. Censoring-motivated MEV opportunities are a result of special interests to not include specific transactions into a block. We focus on SC-enabled opportunities in which transaction execution order matters, as other participants could extract the value, or the mechanics of an application requires an execution before or after a targeted transaction. This case is referred to as transaction-ordering dependent MEV (TOD-MEV) [8].

For transaction ordering, the block proposer is currently the one that can take unfair advantage in exploiting SC-enabled opportunities because of its control of relevant degrees of freedom. Hence, those degrees of freedom need to be reduced. We will base our proposal on the general idea of PBS.

PBS represents a concept to decouple building a block from proposing it and comes in two 'flavors': *out-of-protocol* and *in-protocol*. Currently, MEV-boost [6] represents the dominant out-of-protocol PBS implementation [13]. With MEV-boost, proposers connect to *relays* which offer an auction market for blocks. Builders submit blocks together with a payment for the proposer and proposers simply pick a block to propose with the highest payment. Both proposer and builder trust the relay so that the proposer is not able to steal MEV opportunities within blocks from the builder and the proposer receives the payment offered by the builder. Unsophisticated proposers profit from advanced MEV exploitation techniques with MEV-boost, but relays represent a critical, centralized infrastructure [11]. To curb those re-centralization tendencies, the Ethereum community currently researches in-protocol approaches collectively labeled as ePBS. The idea is to implement PBS at the consensus layer [11]. ePBS aims to provide a 'trustless' infrastructure, removing the need for relays. With protocol-enforced proposer commitments (PEPC), for example, validators can enter verifiable, protocol-related commitments which are enforced by the protocol [10]. In case of block creation, a proposer might sell the protocol-granted right to create and propose a new block or a builder might commit to perform a computing intensive task on behalf of the proposer. As of today, ePBS is not deployed as there remain several complexity and security challenges to be overcome. We think, however, it is likely that ePBS will be deployed in the future, and build our proposal based on it.

Various MEV countermeasures and mitigation strategies have been developed [16]. The authors of [16] present a taxonomy of four categories of countermeasures: *MEV auction platforms* democratize MEV extraction. *Time-based ordering* approaches try to execute transactions in accordance with some fairness criterion with respect to the order of their reception. *Content-agnostic ordering* approaches order transactions independent of their content and typically use some kind of commit-reveal schemes for transactions. *MEV-aware application*s are designed to resist specific types of MEV.

In [13], the idea of a *randomized transaction order* is proposed and [14] presents an extension to a consensus protocol to encrypt transactions and randomize their execution order to mitigate TOD-MEV opportunities. These proposals can be considered as a new form of the content-agnostic ordering category. We follow this line of research and outline a proposal for unpredictable transaction arrangements that is less complex but still effective.

III. TOD-MEV Mitigation: One-To-Three

In this section, we present an approach to mitigate transaction-ordering dependent MEV in Ethereum.

a) Overview: We restrict the degree of freedom to arbitrarily order transactions within a block by separating responsibilities for block creation and proposal across three validators. Therefore, we call the approach One-To-Three (OTT). Essentially, we introduce new consensus rules that enforce a predefined but unpredictable permutation on the selected transactions within a block. In OTT, unpredictability means that no one is able to determine the permutation until a proposer commits to the transactions within the block. By being unpredictable, the resulting transaction order curbs TOD-MEV exploitation as the success of an exploitation attempt is no longer guaranteed. The probability that α ordering-dependent transactions execute in the required order is $\frac{1}{\alpha!} \leq \frac{1}{2}$. This inequality holds because at least two transactions are required for a transaction-ordering dependency and the factorial denotes the number of possible permutations. Therefore, costs for MEV exploitation are increased and successful timing of an exploitation attempt is made less predictable.

b) Separation of responsibilities: In OTT, a valid block contains three signatures from three different validators. Each of these validators is a proposer of the current epoch and performs either the role of a *selector*, *proposer*[1] or *sequencer* for a slot. Proposers are determined by RANDAO as it is currently used and sequencers are selected following the inverse order of the sequence of proposers of the given epoch.[2] The remaining 30 proposers of the epoch act as selectors for each slot. For each slot, selectors, proposers, and sequencers take actions in that order and commit to them via BLS signatures over the result that the next in line verifies. The procedure to create a new block starts with the selectors and is illustrated in Fig. 1. Each selector chooses some transactions from the mempool, e.g. prioritized by the supplied tip, and assembles them into a BLS-signed block which is offered to the proposer. The proposers picks and receives one of the offered blocks together with a valid BLS signature from the selector of the block. The proposer commits to the block by BLS-signing both the block and the signature of the selector. Then, the block and both signatures are handed over to the sequencer who, after validity checks, creates the third signature over the block and the two previous signatures. In addition, the sequencer determines and applies the permutation to the transactions in the block as described in the following, computes the state transition and publishes the final block. To protect the execution related data, e.g. state trie root, during block publishing, the sequencer creates an additional signature which is not required by OTT.

c) Transaction Arrangement: The permutation of a block is derived from the three signatures of the selector, proposer, and sequencer. However, we describe the practical realisation of how to apply the permutation rather than describing the permutation itself. First, the three signatures are XOR'd together into a single value λ. Then, the transactions are sorted according to the XOR distance [9] between their identifying hash and λ. Note that, by depending on λ, the distance calculation and thus the final order of transactions inherits the randomness of λ.

OTT keeps the predetermined permutation ρ unpredictable until the proposer is committed to the transactions of the block. To show this property, we first observe that all three signatures

[1]For the sake of readability, we refrain from delineating between classic PoS proposers and OTT proposers as the latter is a restriction of the former.

[2]As the epoch has an even number of slots, proposer and sequencer do not coincide.

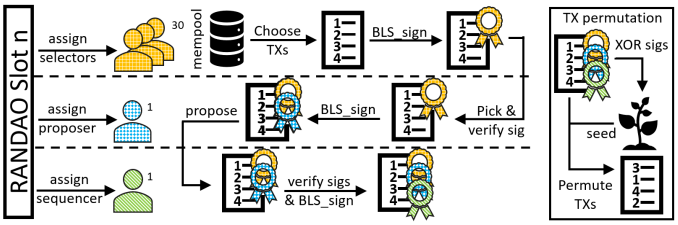

Fig. 1. Visualization of OTT. Left: RANDAO assigns 30 selectors, each may performing the task in the upper swimlane. A single proposer and sequencer is assigned. Right: The transaction permutation is verifiable by all nodes.

are required to determine the final transaction arrangement and thus the permutation ρ. No one except the three validators are able to generate these three signatures as only they know the required private keys. In addition, we observe that the process is strictly sequential. To be accepted by consensus, the signature of the proposer must be valid for the block of the selector together with the signature of the selector. Similarly, the signature of the sequencer must be valid for the block of the selector together with the signature of the selector and the proposer. Due to the first observation, no one is able to determine the permutation ρ without the three signatures, and due to the second observation, the third signature cannot be created without the proposer being committed to the block content by its signature. Thus, the permutation ρ is unpredictable until the proposer is committed to the transactions of the block. Finally, to serve its primary purpose, the permutation ρ must be predetermined so that the three validators cannot choose an arbitrary permutation. First note that the selector sets all fields of a block, except the execution dependent ones like the receipts trie merkle root. Modifying a field invalidates at least the signature of the selector. In addition, the three validators can create exactly one valid signature during the process because of the non-malleability property of the BLS signature scheme and the absence of freedoms to modify the content to be signed. Lastly, the permutation is calculated deterministically based on the three signatures and the transactions in the block. Therefore, the permutation is predetermined. In other words, exactly one permutation is accepted by the consensus rules for a block.

d) Modified Slashing Rules: Until now, OTT has a serious weakness since proposer, sequencer and selector could collude to extract TOD-MEV. For such a collusion attack, the three parties essentially brute-force the whole process with slightly modified blocks as starting points to achieve a suitable transaction arrangement. Even if the proposer equivocates with every repetition, the sequencer is neither forced to publish an equivocation nor has a direct economic advantage in doing so. Worse, the proposer could bribe the sequencer with the selector not to publish the equivocation. Thus, the proposer will not be slashed for the equivocation. To fix this issue, a modification of the equivocation slashing rules is required. Because not just the proposer equivocates in case of a collusion, but also the selector and sequencer, those parties need to be slashable as well. An economic incentive, financed by slashing, is set to report equivocations. A 'whistleblower' who reports an equivocation receives the corresponding Ether as reward. The reward needs to be less than the slashed amount to avoid zero sum games, e.g. in case of self-reporting, but must be greater than a possible gain of bribery through MEV extraction.

IV. Unpredictable Transaction Arrangement: Generalization

The proposed OTT approach can be seen as an instance of a more general class of approaches to achieve unpredictable transaction arrangement (UTA). A UTA-approach establishes an arrangement of transactions within a block that is consensus verifiable, i.e., that is deterministic with only public inputs or zero-knowledge-based, and unpredictable. In this section, we sketch the design space of UTA-based approaches by generalizing OTT and explicating its critical components.

A UTA-instance essentially consists of two parts: *a)* a generator Π for a publicly verifiable (pseudo-)random value s, given a tuple of transactions T, and *b)* a deterministic mapping ω that maps the value s to a unique order $\theta_s(T)$ of the transactions in T such that $\theta_s(T)$ cannot be determined without knowing s. Furthermore, we have to ensure that, when the value s is released by a process, the corresponding $\theta_s(T)$ is released as well. Additionally, for each 'slot' the seed s and thus $\theta_s(T)$ can only be set once.

The deterministic function ω selects the order $\theta_s(T)$ as a permutation of the transaction tuple T. The domain of ω is the cartesian product of the set of all finite sets T of transactions and of the space S of random numbers serving as seed values s. The use of λ and the XOR distance as presented in Section III is obviously just one instance for a resulting transaction order based on a seed value (λ in this case). Regardless of how the transaction set T is permuted based on the seed value s, the function ω must be publicly verifiable. Otherwise, consensus on the final order of transactions, and thus the overall state change of a block, cannot be reached.

The instantiation of the process Π can differ quite significantly, e.g., depending on the underlying consensus system (e.g., PoS or Proof-of-Work (PoW)), and used cryptographic primitives (e.g. signature and encryption schemes). In the following, we first 'deconstruct' OTT presented in the previous section to see how separation of responsibilities and non-malleable signatures lead to the required unpredictability in a PoS-based setting. Then, we briefly sketch how a PoW-based approach provides UTA. Finally, we emphasize the importance of a certain atomicity property: The seed s is determinable as soon as modifying the tuple of transaction T is economically detrimental.

Formally, OTT presumes a set V of entities, each with a slashable security deposit, a non-malleable signature scheme Σ, an oracle Ξ and a canonical identifier γ for blocks. Each entity in V is identified by the public key of its keypair in Σ. The oracle Ξ grants the three rights $R = \{\phi, \chi, \psi\}$ to entities of V. ϕ is the right to select, χ to propose and ψ to sequence. The oracle outputs a set G of grants g_k represented as tuple $\{g_k = (v_k, r) : v_k \in V, r \in R, |\phi| \geq$

$1, |\chi| = 1, |\psi| = 1\}$ for each γ. For simplicity we use the notation v_r to refer to the entity with the granted right r. The process Π is a chained execution of the function Sign of Σ: $\mathrm{Sign}_{\mathrm{v}_\psi}(\gamma, \mathrm{Sign}_{\mathrm{v}_\chi}(\gamma, \mathrm{Sign}_{\mathrm{v}_\phi}(\gamma, \Upsilon)))$. The subscript identifies the entity that performs the function Sign with its private key, γ the canonical block identifier and Υ is a integrity preserving representation of the transaction set T. We assume function Sign to output the generated signature σ and an integrity preserving representation of its inputs.

Let us also sketch an approach for UTA in a PoW context inspired by [14]. In PoW-based blockchains, miners compete with each other to find a nonce so that the hash of the block including the nonce satisfies specific rules. Due to this competition to find the next valid block before anyone else, a successful miner would rather publish their found block including its nonce than to try and repeat the process, thereby satisfying both the unpredictability and atomicity properties of Π. Hence, the nonce can be used and seen as seed s to determine the transaction arrangement.

The sketched PoW approach fulfills the atomicity property as a miner would miss the protocol-defined rewards due to the ongoing competition to find a valid nonce. Note that a nonce is integrally linked to the validity of its block and the transactions it contains, thereby preventing a miner from changing either after completing a proof of work. In a PoS-approach like OTT, the atomicity property is fulfilled too, because the proposer is the sole entity eligible to propose a block and is at risk of being slashed for equivocation if proposing another block with different transactions. With the decision of the proposer for a specific block, all inputs of the remaining deterministic process, i.e. the signature of the sequencer, are fixed even if they are not yet public.

V. Evaluation and Discussion

Unpredictable transaction arrangement increases the risk for an exploiter of TOD-MEV to not only miss the opportunity, but to actually suffer financial loss due to an unexpected order of transactions. The probability of an unexpected transaction order is at least a half. The feature of the proposed OTT-approach is its simplicity based on building on top of the idea of ePBS. We conducted an experiment to measure the computational overhead introduced by OTT, i.e. the signature generation and verification and seed-based transaction permutation. The setup is similar to the visualization in Fig. 1 and we used the BLS12-381 curve, G2-affine signatures and transactions from 1000 Ethereum mainnet blocks. On a laptop, the overhead of the process (per slot) is dominated (90%) by the proposer with less than 2.1s. This is because, in our experiment, the proposer verifies all 30 offered blocks. Consensus nodes need less than 100ms to verify all three signatures of a block and execution nodes permute the transactions in less than 100ms. However, certain open issues and limitations based on the scope of TOD-MEV remain and are discussed in the following. Two limitations, namely brute-force attacks and inability to predict gas consumption, are general issues of TOD-MEV, as outlined by [14], and given for completeness.

a) Brute-force attacks: An attacker submits a high number of transactions, many more than 'minimally required' and thereby significantly increasing the chance of successfully exploiting TOD-MEV also in presence of UTA. However, the costs of the attack increase as well, making at least small value MEV opportunities unprofitable for rationally behaving actors.

b) Gas consumption: With UTA in Ethereum, gas consumption cannot be predicted accurately during block creation. This unpredictability conceals the risk of denial of service attacks. Promising solutions for this issue have been proposed in [14] and can also be applied to OTT/UTA.

c) TEE-supported collusion: OTT collusion-resistance is based on the need to exchange the signatures σ_ϕ, σ_χ and σ_ψ of the selector, proposer, and sequencer to collude. With Trusted Execution Environments (TEEs), it is possible to exchange these signatures confidentially (i.e. encrypted) and compute the seed s and hence the arrangement only inside trusted execution environments. If the TEE outputs a transaction arrangement in which the colluding parties are not able to exploit MEV opportunities in the transaction set T, they can repeat the process Π. Due to the security guarantee of the TEE, the signatures are not revealed to the party operating the TEE, defeating the slashing-based collusion resistance of OTT. However, using a TEE requires some trust, e.g. that the TEE operator does not break the security guarantees [12], [15]. Since operators are incentivized to do so by the whistleblower reward, we assume that validators will not risk their security deposit by using TEEs for collusion.

d) PBS and block dissemination: ePBS is not yet deployed on Ethereum and the block dissemination process might vary depending on the proposal. The block dissemination defines who must (not) have, forward, or publish which data of a block at which time. This dissemination process is security related as, for example, griefing attacks are possible otherwise [3]. We expect OTT to be compatible with most forms of block dissemination as not the block itself must be signed, but just an integrity preserving representation of the transactions within the block.

To justify the required consensus changes of UTA, additional work is warranted, for example: (1) Effects on reordering slippage [1], a measure for increased trading costs in decentralized exchanges, and (2) overall incentive compatibility, especially in the face of off-chain agreement strategies [4] are yet to be examined.

VI. Conclusion

We presented an approach to mitigate TOD-MEV extraction in Ethereum that builds an unpredictable permutation of the transactions before executing them. At its core, OTT relies only on separation of responsibilities and non-malleable signatures to restrict the freedom to arrange transactions. A key feature of the proposed approach is its simplicity and deployability once in-protocol proposer-builder separation is defined. We showed that the underlying idea can be generalized to other blockchain systems as well. However, limitations remain, particularly for the case of TEE-based collusion attacks.

REFERENCES

[1] Austin Adams, Benjamin Y Chan, Sarit Markovich, and Xin Wan. The costs of swapping on decentralized exchanges. *Financial Cryptography (preprint)*, 2024. https://fc24.ifca.ai/preproceedings/206.pdf.

[2] Vitalik Buterin. A Next-Generation Smart Contract and Decentralized Application Platform. Technical report, 2014. https://ethereum.org/en/whitepaper/.

[3] Vitalik Buterin. Proposer/block builder separation-friendly fee market designs, 2021. https://ethresear.ch/t/proposer-block-builder-separation-friendly-fee-market-designs/9725.

[4] Hao Chung, Tim Roughgarden, and Elaine Shi. Collusion-resilience in transaction fee mechanism design. *arXiv preprint arXiv:2402.09321v1*, 2024.

[5] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. pages 1106–1120. IEEE, 2020.

[6] Flashbots. What is MEV-Boost?, 2023. https://docs.flashbots.net/flashbots-mev-boost/introduction.

[7] Lioba Heimbach, Lucianna Kiffer, Christof Ferreira Torres, and Roger Wattenhofer. Ethereum's Proposer-Builder Separation: Promises and Realities. *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 406–420, 2023.

[8] Loi Luu, Duc Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 254–269, 2016.

[9] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the XOR metric. *Lecture Notes in Computer Science*, 2429:53–65, 2002.

[10] Barnabé Monnot. Unbundling PBS: Towards protocol-enforced proposer commitments (PEPC). 2022. https://ethresear.ch/t/unbundling-pbs-towards-protocol-enforced-proposer-commitments-pepc/13879.

[11] Mike Neuder. Why enshrine Proposer-Builder Separation? A viable path to ePBS, 2022. https://ethresear.ch/t/why-enshrine-proposer-builder-separation-a-viable-path-to-epbs/15710.

[12] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. A Survey of Published Attacks on Intel SGX. *arXiv preprint arXiv:2006.13598*, 2020.

[13] Julien Piet, Jaiden Fairoze, and Nicholas Weaver. Extracting godl [sic] from the salt mines: Ethereum miners extracting value. *arXiv preprint arXiv:2203.15930*, 2022.

[14] Julien Piet, Vivek Nair, and Sanjay Subramanian. MEVade: An MEV-resistant blockchain design. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9, 2023.

[15] Michael Schwarz and Daniel Gruss. How trusted execution environments fuel research on microarchitectural attacks. *IEEE Security and Privacy*, 18(5):18–27, 2020.

[16] Sen Yang, Fan Zhang, Ken Huang, Xi Chen, Youwei Yang, and Feng Zhu. SoK: MEV Countermeasures: Theory and Practice. *arXiv preprint arXiv:2212.05111*, 2022.