Breaking the Privacy Barrier: On the Feasibility of Reorganization Attacks on Ethereum Private Transactions

Mengya Zhang
The Ohio State University
Columbus, Ohio, USA
zhang.9407@osu.edu

Xingyu Lyu
University of Massachusetts, Lowell
Lowell, Massachusetts, USA
xingyu_lyu@uml.edu

Jianyu Niu
Southern University of Science and Technology
Shenzhen, CHINA
niujy@sustech.edu.cn

Xiaokuan Zhang
George Mason University
Fairfax, VA, USA
xiaokuan@gmu.edu

Yinqian Zhang
Southern University of Science and Technology
Research Institute of Trustworthy Autonomous Systems
Shenzhen, CHINA
yinqianz@acm.org

Zhiqiang Lin
The Ohio State University
Columbus, Ohio, USA
zlin@cse.ohio-state.edu

***Abstract*—In Ethereum, private transactions are designed to circumvent the public network, but they can sometimes be leaked into the public network before on-chain posting. Motivated by the huge profits of these private transactions, we propose reorganization attacks in the current Proof-of-Stake (PoS) consensus mechanism, enabling malicious validators to actively leak private transactions for profits. While prior research on reorganization attacks has focused on consensus security, our work is the *first* study shedding light on the economic implications of exploiting private transactions. Through theoretical analysis and extensive simulations, we confirm the effectiveness of our attacks. Additionally, we comprehensively examine real-world datasets covering 30,062,232 private transactions from September 15, 2022 to Decemeber 31, 2023 for profit analysis, uncovering that the most lucrative private transactions are often tied to Maximum Extractable Value (MEV). To further bolster the practicability and feasibility of our attacks, we scrutinize real-world cases aligning with our attack patterns. We find that attacks are risk-free due to the predictability of validators' duties. Our findings offer valuable insights into the economics of exploiting private transactions, potential vulnerabilities, and consensus security, laying the foundation for future research.**

***Index Terms*—Blockchain, Private Transaction, Reorganization Attack**

1. Introduction

Private transactions are designed to evade public scrutiny within the Ethereum Peer-to-Peer (P2P) network [1], [2], [3]. The private transactions we mention here belong to transactions on the L1 chain, so the private transactions from other methods (e.g., L2 sidechains) are not in our scope. These transactions are typically processed through private transaction service providers, such as Flashbots [4], which directly transmits the private transactions to validators, concealing them from other nodes in the P2P network to ensure privacy. However, privacy comes with a cost, as users need to pay higher transaction fees or make direct payments to validators as service fees. As a result, private transactions can yield substantial profits for validators. For example, one documented private transaction [5] produced a profit of 720.26 *ETH*, equating to nearly one million USD.

While the goal of private transactions is to avoid appearance in the public P2P network, we have detected that a fraction of private transactions can be leaked into the public P2P network, negating their intended privacy, since every Ethereum node can observe these leaked private transactions (contradicting the purpose of utilizing private transactions). To verify this leakage, we deployed Ethereum nodes across two continents, and identified an average leakage rate of 0.51% over a 14-day span (see Appendix A). For instance, a specific private transaction [6] reached our local nodes and was mined by the validator *Lido* [7] less than twenty seconds later. The unintended exposure of private transactions can cause significant financial harm on honest validators who mine them, while simultaneously benefiting other validators who mine these transactions. For instance, the profitable private transaction mentioned above, which yielded 720.26 *ETH* in profits, could incur severe financial harm if leaked, depriving its validator of these earnings.

Although many existing works [8], [9], [10], [11], [12], [3] have focused on quantifying MEV or assessing the characteristics of private transactions, the attacker's perspective in actively inducing leaks of private transactions and profiting from them has been explored. For instance, Lyu et al. [3] noted that private transactions could leak, but they only measured the leakage without considering potential attacks. In addition, most studies focus on private transactions in Proof-of-Work (PoW) Ethereum,

while ignoring the current Proof-of-Stake (PoS) Ethereum. Unlike these works, we dive into the leakage attacks of private transactions in both PoW and PoS Ethereum. Our investigation indicates that these leaks are primarily due to block reorganizations, which often occur inadvertently. For example, a late-arriving proposed block may be reorganized, causing all contained private transactions to be leaked. Although most reorganizations probably appear accidental (due to network latency), the possibility of deliberate attacks persists. This scenario prompts the key questions that this paper seeks to answer: *How to deliberately (or actively) leak private transactions, and how to make profits by abusing the leaked private transactions?*

To bridge this critical gap, we thoroughly explore attack strategies to actively leak private transactions for profits in Ethereum. Our examination focuses on two types of attacks: *retrospective* and *prospective* reorganization attacks, covering all possible reorganization scenarios to achieve our goals. Specifically, our attacks leverage known consensus issues (reorganization issues), not directly targeting the safety of consensus. Reorganization attacks are not new; they were originally mentioned in the Bitcoin white paper in 2008 [13] and have been used in various attacks ever since. Unlike existing reorganization attacks to disrupt consensus security, our *retrospective* and *prospective* attacks focus on posted and upcoming private transactions for profits, respectively, expanding existing ex post and ex ante reorganization strategies [14]. Specifically, what distinguishes our proposed attacks from existing ones is the nature of private transactions, *i.e.*, they are visible only to specific validators. We find that executing a *retrospective* reorganization attack in PoS Ethereum is extremely challenging and almost impractical. In contrast, the *prospective* reorganization attack turns out to be more feasible, supported by our analysis, simulation and measurement of real-world datasets.

Our study provides comprehensive analysis of the attack model, strategy, staking requirements, and the minimum private transaction profits needed to incentivize attacks. Through simulation experiments, we validate the expected success rates of our proposed attacks derived from our theoretical analysis. To assess the profits of private transactions and identify real-world attack cases that align with our attack patterns, we analyze real-world data spanning 15.5 months in PoS Ethereum (from September 15, 2022, when Ethereum transitions from PoW to PoS, to December 31, 2023).

Contributions. We make the following contributions:

- **Novel attacks.** We propose two types of reorganization attacks for leaking private transactions and stealing their profits: *retrospective* and *prospective*; we analyze the attack model, strategy, and requirements for attackers, including required staking and profit. We find that executing a *retrospective* attack is nearly impossible (e.g., requiring at least 66.67% staking for attackers, whereas a *prospective* attack becomes feasible with a small percentage of staking controlled by attackers, such as 1%, resulting in a 21.15% attack success rate. Additionally, we investigate the *prospective* reorganization attack under the proposer boost mechanism (§2.2) that boosts the attestation votes for timely proposed blocks, which raises the staking requirement for an attack. Despite this, we find it remains feasible, with a 3.34% staking resulting in a 92.49% success rate when the proposer boost score is 90%. Moreover, we conduct thorough simulation experiments to prove the attack success rate aligns with our theoretical analysis.
- **Real-world analysis of profits and attacks.** We explore potential factors inducing reorganization attacks, especially MEV related private transactions. We find that all the top 10 most profitable private transactions are associated with MEV, and the highest-grossing private transaction [15] yields 541.90 *ETH*, equivalent to approximately 1.3 million dollars as of January 1, 2024. Specifically, 23.57% MEV related private transactions in our 15.5-month dataset can provide enough profits to motivate our attacks; these profits exceed the average profit of all private transactions (0.01 *ETH*). We extend our analysis to real-world data, identifying instances that align with our attack patterns. This validation confirms the feasibility of our proposed attacks, allowing us to delve into the potential gains for attackers. In our investigation, we pinpoint 124 potential instances of prospective reorganization attacks, wherein the average profits from private transactions amount to 0.23 *ETH* — notably surpassing the average profits across all blocks (0.13 *ETH*).
- **Open-source datasets and scripts.** To encourage future research, our datasets and scripts are publicly accessible at https://anonymous.4open.science/r/BreakingPtxs-4A2B/. We believe that this study can contribute valuable insights into the security of Ethereum's private transactions, aiding in the creation of a fortified blockchain ecosystem.

2. Preliminaries

In this section, we present the essential foundations, covering Ethereum basics (§2.1), consensus mechanism — PoS (§2.2), and transaction mining process (§2.3).

2.1. Ethereum Basics

Peer-to-peer (P2P) network. Ethereum [16] operates as a public, decentralized, and permissionless blockchain platform. In Ethereum, nodes, often referred to as Ethereum clients, connect through a P2P network for efficient data communication, including transaction broadcasting. This decentralized network architecture ensures that no single entity has control over the entire network, fostering resilience against censorship and single points of failure. Participants in the network, whether individuals or organizations, can interact directly with each other without the need for intermediaries, enabling a truly peer-to-peer exchange of value and information.

Transactions. Ethereum has two types of accounts: Externally Owned Accounts (EOAs) and Smart Contracts. EOAs function as standard cryptocurrency wallets without code, while smart contracts are Turing-complete, self-executing programs that run on the Ethereum blockchain. Transactions always originate from an EOA and are directed towards a specific account. If the receiving account is also an EOA, the transaction represents a straightforward transfer between two accounts. However, if the receiving account is a smart contract, the Ethereum Virtual Machine (EVM) executes the related contract.

Transaction cost. Every transaction in Ethereum is required to pay a transaction fee, which is calculated as $TxFee = UsedGas \times GasPrice$, where *UsedGas* is the amount of gas used for executing a transaction and *GasPrice* is the amount the user is willing to pay per unit of gas. According to EIP-1559 [17], *GasPrice* is calculated as: $GasPrice = Basefee + PriorityFee$, where *Basefee* is the required fee determined in each block and will be burnt later. Specifically, the gas fee must be paid using *ETH*, which is the native token of Ethereum. Aside from transaction fees, transactions may involve a direct payment to block creators in the form of tips. Generally, the more substantial the tips offered by transactions, the higher the likelihood of their swift inclusion in desired positions in the mining process.

2.2. Proof-of-Stake (PoS)

Epochs, slots and stages. To reduce the energy waste, Ethereum switched its consensus mechanism from PoW to PoS [18] on September 15, 2022. In PoS Ethereum, time is organized into epochs, each lasting 6.4 minutes and consisting of 32 slots, each of 12 seconds duration. At the onset of each epoch, validators, virtual entities tasked with participating in the consensus process by proposing and validating transaction blocks, are shuffled into committees using a randomness source known as RANDAO [19]. Within each slot, one validator is chosen as the block proposer, while the remaining validators serve as attestation validators.

Each slot is further divided into three stages, each lasting four seconds. The stages include: *1) Stage 1: block proposal.* In this stage, the assigned proposer presents a block to the network. *2) Stage 2: attestation.* In this stage, attestation is performed to verify the block. When attestation validators observe the proposed block or the first stage concludes, they are required to vote on its validity. *3) Stage 3: aggregation.* In the final stage, aggregators combine the individual attestations from the validators. They create a single signature by merging the signatures of all attestation validators.

Specifically, there is no penalty for proposing a late block or even failing to propose a block. Theoretically, a block should be proposed within 12 seconds since there is a 12-second interval between every two blocks. In fact, a block is encouraged to be proposed in the first 4 seconds so that 8 seconds are left for validators' attestation. However, a block proposer can release as late as 11 seconds into the slot and still be considered canonical. A slot can be divided into three phases [20]: *1)* At the beginning of the slot, a block proposer is supposed to propose a block. *2)* Committee members are supposed to attest when they hear a valid block or 4 seconds into the slot—whichever comes first. *3)* Aggregators are supposed to release their attestations 8 seconds into the slot.

Fork choice. Fork choice is the crucial process of determining the valid blockchain when validators hold differing views of the latest block. In PoS Ethereum, the Latest Message Driven Greediest Heaviest Observed SubTree (LMD-GHOST) algorithm [21] is employed to achieve consensus and select the canonical chain based on the weight of validators' attestations. This algorithm prioritizes the chain with the highest weight of attestations as the canonical chain. To illustrate, if each attestation from a validator carries an equal weight value of 1, the chain with the most attestations from validators becomes the canonical chain. This mechanism ensures the integrity and coherence of the blockchain despite potential divergent views among validators.

Proposer boost mechanism. While the existing fork-choice protocol derived from LMD-GHOST provides a robust basis for consensus, it remains vulnerable to reorganization attacks. In response to these vulnerabilities, the proposer boost mechanism [22] has been introduced. This innovative mechanism aims to bolster the security of the protocol by granting additional fork-choice vote weights to timely proposals during their slots. By doing so, it fortifies the blockchain against reorganization attacks that lack sufficient saved attestations to surpass the weighted threshold. Notably, the proposer boost mechanism is an optional feature for validators and the related score is adjustable, providing them with flexibility in adapting their strategies to mitigate potential threats.

Rewards, penalties, and slashings. Validators obtain rewards for attesting and proposing blocks. Additionally, they receive rewards for serving as a sync committee members in each epoch. They will also be penalized for making incorrect attestations and being inactive. Attestation rewards are calculated based on three factors: the source block, the target block, and the head block. If any information is found to be inaccurate, the attestation will be deemed incorrect, and the attestation validator will not be rewarded accordingly. Moreover, validators will face more severe punishments if *slashing* conditions are met, including producing two blocks for the same slot or voting for two blocks in the same slot.

Proposer-builder separation (PBS). PBS splits the tasks for validators, who serve as block creators in PoS Ethereum. Under the PBS model, block builders are tasked with the construction of blocks, while validators assume the role of proposing these blocks. This separation ensures a clear delineation of duties: block builders focus on the technical aspects of block construction, while validators, also known as proposers when selected to propose blocks, are responsible for advancing blocks for consensus consideration. Notably, while PBS has been adopted, it

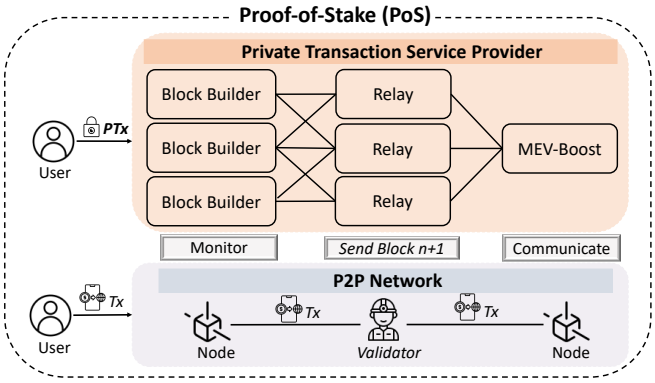

Figure 1: The process of mining a public transaction (Tx) vs. a private transaction (PTx) in PoS Ethereum. Tx is sent into the P2P network, while PTx is directly sent to validators via private transaction service providers.

remains optional rather than a strict requirement, providing flexibility within the ecosystem to accommodate diverse operational preferences. Also, some validators still build blocks even with the existence of PBS.

2.3. Transaction Mining Process

Public transactions vs. private transactions. Private transactions are crafted to circumvent the public P2P network, whereas public transactions traverse through the P2P network. Fig. 1 shows the workflow of mining a public transaction (Tx) vs. a private transaction (PTx), where there is a key difference. Public transactions are sent over the P2P network and broadcast to all nodes in the network. Validators in PoS will receive these transactions as well. However, private transactions are sent over a private transaction service provider.

Private transaction mining. Due to the widely adopted PBS (§2.2), block builders primarily receive private transactions and construct blocks, though validators can still take on this role occasionally. Block builders bundle private and public transactions into newly mined blocks and forward the optimal block to relays, which then send the best block to MEV-Boost run by validators. The proposer receives and signs only the block header before proposing it to the network.

3. Overview

In this section, we discuss the problem statement (§3.1), threat model (§3.2), and overview of our reorganization attacks (§3.3).

3.1. Problem Statement

The problem. Our primary objective is to leak private transactions for gaining profits through consensus-level attacks. This objective arises from two key observations:

Leakage of private transactions. While private transactions aim to evade exposure on the public P2P network, our investigation reveals that a portion of them can be inadvertently leaked into the public domain. A private transaction is considered leaked when it is submitted to circumvent initial exposure on the public P2P network but is subsequently disclosed in the public network before being successfully mined. We cover all types of private transactions, regardless of the methods used to send them or create them. To be more specific, our scope is not limited to attacks involving MEV-Boost, since MEV-Boost is merely one of the methods for creating private transactions.

Profitability of private transactions. Our investigation also reveals that private transactions typically yield greater profits compared to normal transactions. These profits, earned by validators (or block proposers) who mine transactions, consist of direct payments from users and tips received from transaction fees after deducting the burnt fee [17]. Moreover, we do not consider potential offline payments to attackers, which is unlikely and beyond our scope.

Problem definition. The alignment of our objective with the lucrative nature of private transactions underscores the significance of the attacks. We focus on two critical aspects:

Attackers - malicious validators. Our investigation centers on malicious validators with the potential to deliberately disclose private transactions and subsequently capitalize on these disclosures to maximize profits. The attackers possess the capability to participate in block voting and construct new blocks specifically aimed at mining lucrative private transactions. While other methods, like copycat attacks [23], may also exploit leaked transactions, they are beyond the scope of our research. Moreover, we do not consider the bribery attacks [24], [25], [26], in which attackers can collude with honest validators to launch attacks.

Attack methods - reorganization attacks. The leakage of private transactions necessitates attackers to execute attacks before block finality is achieved. Note that we have deployed Ethereum nodes and identified this leakage (see Appendix A). Finality denotes the assurance that a block within the blockchain cannot be modified or expunged without the requirement of burning at least 33% of the total staked *ETH* in the whole Ethereum network. Since attaining the 33% threshold is notably challenging, it is generally accepted that once finality is reached, a block is typically immutable. Therefore, the most effective method entails launching reorganization attacks to reorganize blocks containing lucrative private transactions. To achieve this, we explore reorganization attacks engineered to violate the finality of private transactions. Our research comprehensively delves into two types of reorganization attacks, namely *retrospective* and *prospective*, for leaking private transactions and deriving profits from these leaks.

3.2. Threat Model

We establish a threat model wherein malicious validators can utilize blockchain consensus mechanisms to execute consensus-level attacks, specifically reorganization attacks.

Attack	Model		Strategy	Requirements				
	#Honest Block	#Attack Block		Vote	Proposer	Staking, Success Rate	Hashrate, Success Rate	Profits (*ETH*)
Retrospective	1	1	Target honest and steal	Eq. (14)	slot n+2	>= 66.67%, 100%	N/A	> average
Prospective	1	1	Hide, release, and steal	Eq. (1)	slot n+1 and n+3	1%, 21.15% (Fig. 6)	N/A	> average
Prospective - Proposer Boost	1	1	Hide, release, and steal	Eq. (10)	slot n+1 and n+3	3.34%, 92.49% when PB = 90% (Table 3)	N/A	> average

TABLE 1: Summary of our *Retrospective* and *Prospective* reorganization attacks. *Profits (ETH)* have dictated the necessary gains, necessitating profits above the average from private transactions, irrespective of the controlled staking percentage.

Assumptions. To provide estimated yet realistic requirements for successfully executing our reorganization attacks, we have the following assumptions:

- *Economically rational validators.* We assume validators, excluding attackers, behave in an economically rational manner. Likewise, attackers make decisions driven by the incentive of gaining profits from private transactions.
- *Standardized vote weights.* Every validator, including attackers, has the vote right during the mining process. We do not consider the varying weights of individual validators; instead, we standardize the weight of each validator's vote to 1, similar to existing work [27], [14].
- *Single honest block attack strategy.* We assume that attackers will attempt to fork out one honest block rather than multiple honest blocks, given the low probability of multi-block attacks.
- *Economically bounded attackers.* We consider attackers as economically restricted validators, implying limited staking for participation in the mining process. Consequently, attackers can only control a fraction of the total validators within the Ethereum network.
- *Attacker capabilities.* Attackers cannot predict the contents or profits of upcoming blocks. They can only delay proposing their attack blocks for a few seconds within a 12-second window. Additionally, they can vote for a fork when necessary and, if chosen as proposers, they can mine and rearrange transactions in blocks.
- *Attack strategies.* We do not assume that validators or attackers will propose a specific block. Attackers can propose any block to initiate an attack, provided they have sufficient staking in PoS Ethereum to control enough validators for attestation.
- *Attack profits.* We assume attackers adjust the private transactions in their blocks to maximize their profits, taking into consideration the practical limitation of block space. Additionally, we calculate the rewards or penalties for attacks without considering exceptional cases such as slashing, as they occur infrequently.
- *Uniform proposer boost score.* While not all validators may adopt the proposer boost mechanism, in instances where it is utilized, we assume that all participating validators possess identical proposer boost scores.

3.3. Reorganization Attacks Overview

We propose two types of reorganization attacks, namely *retrospective* and *prospective*, for leaking private transactions and gaining profits from these leaked transactions at the expense of honest validators. In a successful attack, all private transactions in the reorganized honest block are exposed, but attackers may only mine some leaked profitable private transactions to maximize their profits. These attacks encompass various aspects, including the attack model, strategy, staking requirements, and the minimum profit needed to incentivize the attacks. A summary of these aspects can be found in Table 1, and symbols used throughout the attack analysis are listed in Table 2. Specifically, the *retrospective* reorganization attack targets profitable private transactions observed in an honest block, while the *prospective* reorganization attack focuses on upcoming private transactions in a future honest block.

Symbol	Description
m	block number
PTx-i	private transaction with i profits
P_{one}	the probability of successfully launching the attack
P	the success rate of launching at least one attack per day
R_{NA}	the total reward for attackers without attack
R_A	the total reward for attackers with attack
R_{AF}	the total reward for attackers when attack fails
R_{ptx}	the stolen profits of private transactions from honest blocks
R_{B_i}	the total reward for a block proposed in slot i without attack
R'_{B_i}	the total reward for a block proposed in slot i with attack
n	slot index
M_1	attacker controlled attestation validator count in slot n+1
M_2	attacker controlled attestation validator count in slot n+2
N	the count of attestation validator per slot
p	the attacker controlled staking percentage
P_v	the probability of satisfying vote requirement
P_p	the probability of satisfying proposer requirement
C	the chances of launching attacks per day
R_V	attestation reward

TABLE 2: Summary of symbols used by our two types of reorganization attacks.

4. Reorganization Attacks in PoS Ethereum

We propose two distinct types of reorganization attacks in PoS Ethereum, namely *retrospective* and *prospective* attacks, to leak private transactions and steal profits from them. Since *retrospective* attack (Appendix B) is very challenging to execute and requires at least 66.67% staking. In this section, we focus on *prospective* attack. We present the attack model, examining factors including strategy, required attacker controlled staking, and needed private transactions profits for motivating *prospective* attacks (§4.1). We also examine *prospective* reorganization attacks under the proposer boost mechanism (§4.2). Additionally, to confirm that our proposed attacks can be launched with the expected success rate, we perform thorough simulation experiments (Appendix C).

4.1. Prospective Reorganization Attack

Attack model. Fig. 2 illustrates the *prospective* reorganization attack, given three adjacent slots within the same epoch.

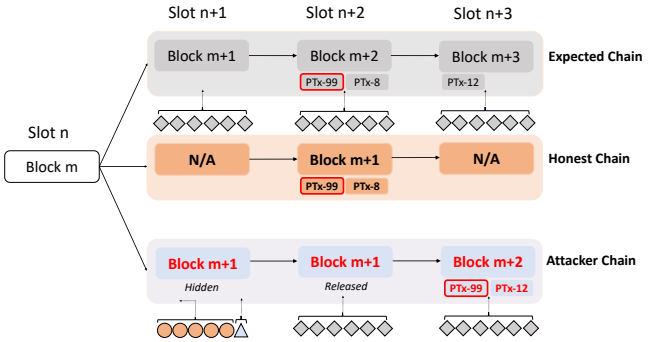

Figure 2: The *prospective* reorganization attack incentivized by private transactions, ending with leaking private transactions PTx-99 and PTx-8 and stealing the profits of PTx-99 from the honest block proposer. *PTx-i* represents the private transaction with profits i. ◇ represents a vote by either attacker or honest validators; ○ represents a vote in the block of the honest chain controlled by honest validators; △ represents a vote in the block of the attacker chain controlled by attackers. All the honest validators in slot $n+1$ vote for block m in slot n.

In PoS Ethereum, each slot has an assigned proposer for proposing one block. Every proposed block might contain a few private transactions whose profits will be obtained by the block proposer mining this block. We assume block m+1 in slot $n+1$ does not have profitable private transactions, whereas block m+2 proposed in slot $n+2$ has the private transaction PTx-99 with 99 profits and the private transaction PTx-8 with 8 profits, and block m+3 proposed in slot $n+3$ has the private transaction PTx-12 with 12 profit. We further assume attackers are selected to propose the block in slot $n+1$ and slot $n+3$, while honest validators propose the block in slot $n+2$. To be specific, attackers here represent a group of malicious validators. Without attacks, the blockchain will remain as the expected chain, where typically one block is proposed at one slot and private transactions will be included into the related block. However, attackers may launch a *prospective* reorganization attack driven by the huge profits of private transactions (e.g., PTx-99) from the block m+2, proposed in slot $n+2$ by honest validators.

Attack strategy. The attackers deliberately construct a fork chain with the following three steps.

Step 1: hide attack block in slot n+1. In slot $n+1$, attackers propose attack block m+1 in slot $n+1$ but hide the block. We assume attackers control M_1 votes, then honest validators control N-M_1 votes, where N represents the number of committees for votes per slot. Since the attack block m+1 is hidden, N-M_1 honest validators cannot obtain the block and then vote for the block m in slot n. Moreover, the M_1 will vote for the attack block $m+1$ for later use.

Step 2: release attack block in slot n+2. In slot $n+2$, the honest proposer of slot $n+2$ appends its block m+1 to block m in slot n, since it has no view of block in slot $n+1$. At this time, attackers then release the hidden attack block m+1 at almost the same time as the honest block m+1 in slot $n+2$ is proposed by honest proposer. Thus, attestation validators in slot n+2 have two views: the honest chain whose head is slot $n+2$ and the attacker chain whose head is slot $n+1$. To succeed in the attack, attacker chain should obtain more votes from attestation validators than honest chain. Since the current vote for honest chain is 0, the attack can succeed as long as there is at least 1 validator voting for the attacker chain before. Therefore, if M_1 is no less than 0, the attack will succeed.

Step 3: steal profits in slot n+3. In slot $n+3$, all the validators in slot $n+3$ vote for the attacker chain since the attack has succeeded. Moreover, the attackers pick the most profitable private transactions from reorganized honest blocks and steal their profits.

Attack example. As shown in Fig. 2, attackers need to control at least M_1 = 1 vote in slot $n+1$ and attach the vote to the hidden attack block, whereas the left $N - M_1$ = 6 - 1 = 5 honest validators in slot $n+1$ vote for the block m in slot n since they have no view of proposed block in slot $n+1$. Then attackers release attack block m+1 when honest block m+1 is proposed in slot $n+2$. At that time, the attacker chain has already obtained more votes than the honest chain, since the honest chain has 0 vote and the attacker chain has 1 vote. Therefore, the attack succeeds and all the validators in slot $n+2$ will vote for the attacker block m+1 in slot n+2. Thus, the honest block m+1 in slot n+2 is reorganized. As the attack succeeds, PTx-99 and PTx-8 are leaked into the public P2P network, rendering them no longer private. Consequently, attackers can pick any private transactions from the set of PTx-99, PTx-12, and PTx-8 for mining into block m+2 in the attacker chain. Intuitively, attackers will mine as many as private transactions they can to maximize their profits. However, assume there are conflicts between PTx-12 and PTx-8 (e.g., two MEV searchers targeting at the same MEV opportunity), PTx-12 will be mined and PTx-8 will be discarded since PTx-12 has more profits. Therefore, while both PTx-99 and PTx-8 are leaked, only the profits of PTx-99 are stolen by attackers.

Attack staking requirement and success rate. In accordance with the attack strategy, the staking requirement for the attack is as follows:

$$M_1 > 1. \tag{1}$$

Assume p denotes the percentage of staking controlled by attackers, signifying that attackers have control over p of all active validators. Therefore, the probability of an attacker being chosen as the attestation validator in a specific slot is p, and the probability of satisfying the vote requirement under p is as follows:

$$P_v = 1 - (1-p)^N. \tag{2}$$

Moreover, attackers need to be proposers of both slot $n+1$ and $n+3$ and honest validators should be proposer of the slot $n+2$. Therefore, the probability of satisfying the proposer requirement should be as follows:

$$P_p = p^2(1-p). \tag{3}$$

Therefore, the success rate (P_{one}) of one attack given three adjacent slots under p should be the probability of satisfying both vote and proposer requirements, as follows:

$$P_{one} = P_v \times P_p. \tag{4}$$

Assume C represents the number of attacks an attacker can launch per day. The lower bound of C will be 2,400, since there are 7,200 slots per day and one attack requires three adjacent slots. The higher bound of C will be 6,750 with 30 chances per epoch multiplied by 225 epochs. However, we will take the lower bound as the value of C (2,400), since it is more realistic. Therefore, the success rate (P_s) of launching at least one successful attack per day is as follows:

$$P_s = 1 - (1 - P_{one})^C. \tag{5}$$

Attack profits requirement. In PoS Ethereum, the proposer and attestation validators for each slot are determined in advance, for a minimum of one epoch [28]. Thus, attackers know whether the attack requirements are satisfied before launching the attack. If the attack requirements are not met, attackers will act honestly; otherwise, they will proceed with the attack. Hence, as long as the profits attained by attackers when the attack is successful (R_{AS}) surpass those in a non-attack scenario (R_{NA}), attackers will launch attacks to maximize their gains.

Assume M_1 represents the attacker controlled attestation validator number in slot $n+1$, R_V is the attestation fee, and R_{B_i} is the total reward for a block proposed in slot i. The total reward R_{NA} for the attacker without launching an attack is:

$$R_{NA} = M_1 \times R_V + R_{B_{n+1}} + R_{B_{n+3}}. \tag{6}$$

The total reward R_{AS} for the attacker successfully launching an attack is as follows, where R_{ptx} represents the profits of private transactions from honest block stolen by attackers (e.g., profits from the private transaction PTx-99):

$$R_{AS} = M_1 \times R_V + R'_{B_{n+1}} + R'_{B_{n+3}} + R_{ptx}. \tag{7}$$

As R_{ptx} consistently exceeds or equals 0, indicating that R_{AS} is always greater than or equal to R_{NA} at any given moment, a *prospective* reorganization attack does not hinge on any private transaction profits. However, we suspect that despite meeting the attack staking requirement, the attack might not be executed if the profits from potentially leaked private transactions are not enough. The reason is that there might be private transactions in the original attack blocks that could yield profits to attackers even without executing attacks; only when the profits from potentially leaked private transactions surpass a threshold such as the average profits of all private transactions, will the attacks be launched (see §5.1.3 for real-world data analysis).

Finding 1: The likelihood of a *retrospective* reorganization attack is minimal, while executing a *prospective* reorganization attack is relatively straightforward, with easily attainable required profits from private transactions.

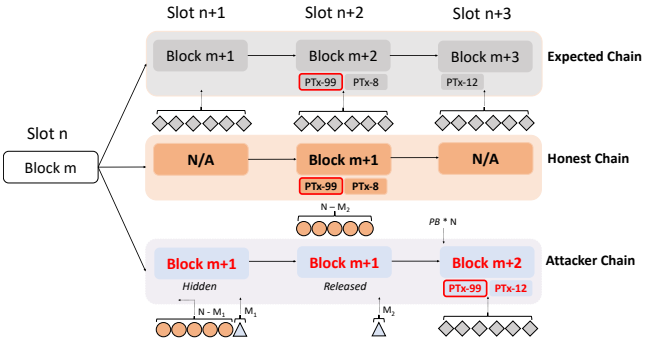

Figure 3: The *prospective* reorganization attack incentivized by private transactions with *proposer boost* mechanism, ending with leaking private transactions PTx-99 and PTx-8 and stealing the profits of PTx-99 from the honest block proposer. *PTx-i* represents the private transaction with profits i. ◇ represents a vote by either attacker or honest validators; ○ represents a vote in the block of the honest chain controlled by honest validators; △ represents a vote in the block of the attacker chain controlled by attackers. All the honest validators in slot n+1 vote for block m in slot n.

4.2. Prospective - Proposer Boost

The current fork-choice protocol is susceptible to our proposed reorganization attack, which demonstrates success with control over just one vote; hence, the proposer boost mechanism is introduced for mitigation (§2.2). Assuming a proposer boost score of *PB*, this mechanism enhances the timely block with an equivalent of $PB \times N$ votes. If a block proposed in slot n is received within the first four seconds of the slot, it benefits from the boost throughout the duration of slot n. Once slot n concludes, the boost is revoked, and subsequent vote weight calculations rely solely on attestations.

Attack strategy. While the attack model remains consistent between the one with the proposer boost mechanism and the one without it, the attack strategy is slightly different, as shown in Fig. 3.

Step 1: hide attack block in slot n+1. In slot $n+1$, attackers propose and hide attack block m+1 in slot $n+1$. Assuming attackers control M_1 votes, N-M_1 honest validators vote for block m in slot n, while M_1 attacker validators vote for the hidden attack block $m+1$.

Step 2: release attack block in slot n+2. In slot $n+2$, the honest proposer of slot $n+2$ appends block m+1 to block m in slot n, and the attackers release the previously hidden attack block m+1. Utilizing the proposer boost mechanism, the honest chain accumulates $PB \times N$ votes due to the timely proposal of honest block m+1. If $PB \times N$ surpasses the attacker votes M_1, the honest block m+1 attaches to block m, and the attack is unsuccessful. Ultimately, M_2 validators, assuming attackers control M_2 votes, vote for the attacker chain, while $N - M_2$ honest validators vote for the honest chain. Notably, the $PB \times N$ votes for the honest chain no longer exisit after slot n+2.

Step 3: steal profits in slot n+3. In slot $n+3$, the attacker chain has gained M_1 votes in slot $n+1$, M_2 in slot $n+2$, and $PB \times N$ in slot $n+3$ due to the proposer boost

PB	10%	20%	30%	40%	50%	60%	70%	80%	90%
p	30.01%	26.67%	23.34%	20.01%	16.67%	13.34%	10.01%	6.67%	3.34%
P_s	1	1	1	1	1	1	1	1	0.9249

TABLE 3: The required staking percentage (p) for a successful *prospective* reorganization attack and the probability of launching at least one successful attack per day (P_s) considering the proposer boost score (PB).

mechanism. For the attack to succeed, the attacker chain must secure more votes than the honest chain, equivalent to $N - M_2$. Consequently, attackers can pilfer profits from private transactions in reorganized honest blocks by incorporating such transactions into attacker block m+2 in slot $n + 3$.

Attack staking requirement and success rate. In accordance with the attack strategy, the staking requirement for the attack is as follows:

$$PB \times N > M_1. \tag{8}$$

$$M_1 + M_2 + PB \times N > N - M_2. \tag{9}$$

Considering that $M_1 \approx M_2 \approx p \times N$, where p represents the attacker controlled staking percentage, the staking requirement translates to:

$$P_v : PB > p > (1 - PB)/3. \tag{10}$$

Specifically, if p exceeds PB, the *prospective* reorganization attack can succeed preemptively. However, as p typically falls below PB, we can disregard the upper limit of p. Furthermore, attackers must act as proposers for both slot $n+1$ and $n+3$, while honest validators should be proposers for slot $n + 2$. Therefore, the probability of meeting the proposer requirement aligns with Equation (3). Similarly, the success rate (P_{one}) of launching one attack given three adjacent slots under p should be the probability of satisfying both vote and proposer requirements, equivalent to Equation (4). Additionally, the success rate (P_s) of launching at least one successful attack per day corresponds to Equation (5).

The value of the proposer boost score (PB) is dynamic, initially set at 70% upon its implementation on November 23, 2021, later adjusted to 40% on May 20, 2022 [29], and has remained at 40% ever since. To access the impacts of PB, Table 3 illustrates the necessary staking percentage (p) and the probability (P_s) under varying proposer boost score (PB). Generally, as PB increases, both p and P_s decrease. Notably, when PB ranges from 40% to 80%, p remains sufficiently large, and P_s consistently equals 1.0. This indicates a consistent likelihood of at least one *prospective* reorganization attack occurring per day within this range of PB.

Attack profits requirement. The total reward R_{NA} for the attacker without launching an attack and the total reward R_{AS} for the attacker successfully launching an attack are as follows:

$$R_{NA} = (M_1 + M_2) \times R_V + R_{B_{n+1}} + R_{B_{n+3}}. \tag{11}$$

$$R_{AS} = (M_1 + M_2) \times R_V + R'_{B_{n+1}} + R'_{B_{n+3}} + R_{ptx}. \tag{12}$$

R_{AS} is always greater than or equal to R_{NA} at any given moment given $R_{ptx} >= 0$. Likewise, the attacks will only be initiated if the profits from potentially leaked private transactions exceed a threshold, such as the average profits of all private transactions (refer to §5.1.3 for real-world data analysis).

Finding 2: The prospective reorganization attack gains feasibility as the proposer's boost score increases relative to the staking requirement, while the profit requirement remains constant and easily achievable.

5. Real-world Analysis in PoS Ethereum

In this section, we conduct a thorough real-world analysis of private transactions spanning 15.5 months from September 15, 2022 to December 31, 2023, focusing on profits (§5.1) and attack analysis (§5.2).

5.1. Profit Analysis

5.1.1. Datasets. To gain a comprehensive understanding of private transaction profits, we collect a large-scale PoS dataset, starting from September 15, 2022 (block 15,537,394) to December 31, 2023 (block 18,908,894), containing a total of 499,327,807 transactions in 3,371,501 blocks. To collect the necessary data, we use various reliable sources to collect the following information:

- **Transaction information.** We collect 7-tuple *{Transaction Hash, Block Number, Sender, Receiver, Value, Input, Index}* for every transaction from our *Geth* [30] node, which is an official Ethereum client implemented in Golang language. We also 3-tuple *{Transaction Hash, Used Gas, Gas Price}* for every transaction from *EthereumETL* [31], which is an open source tool to provide Ethereum on-chain data.
- **Private transaction label.** To detect private transactions, we leverage the reliable open-source datasets provided by Blocknative's mempool data [32] (see §7 for discussion). We identify 30,062,232 private transactions, constituting 6.02% of the total transactions.
- **Block information.** We collect 8-tuple *{Block Number, Timestamp, Proposer, Block Reward, Gas Limit, Gas Used, Basefee, Burnt Fee}* for every block from *Geth*.

5.1.2. Blocks. In our 15.5-month PoS dataset, there are 3,371,501 blocks in total, and 2,500,979 of them (74.18%) contain private transactions. The average profits of private transactions per block in these cases amount to 0.13 *ETH*, representing the potential gains for attackers using our proposed reorganization attacks. Moreover, Fig. 4 illustrates 2,479,101 (99.13%) blocks with private transaction profits not exceeding 1 *ETH*, while the remaining 21,878 (0.87%) blocks exhibit profits surpassing 1 *ETH*. Specifically, 744,770 blocks (29.78%) have profits of at least the average value of profits (0.13 *ETH*), which can offer sufficient profits of private transactions to motivate our *prospective* reorganization attacks.

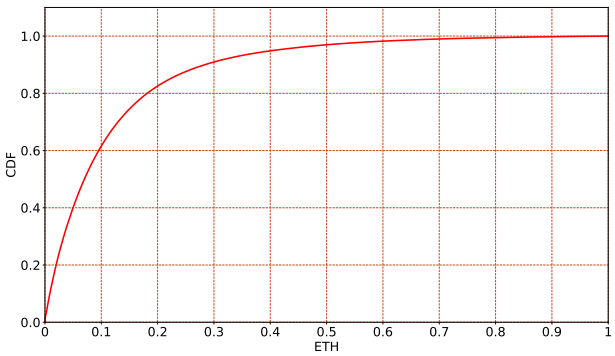

Figure 4: The CDF of profits from private transaction to validators per block in our 15.5-month dataset.

Hash	Profits (*ETH*)	Block	Validator	MEV
0xa954b21c [15]	541.90 (78.32%)	16,867,030	0xdafea492 [33]	Arbitrage
0x0bff9cfa [34]	337.91 (99.97%)	15,802,413	0xee5f5c53 [35]	Arbitrage
0xe336debd [36]	195.88 (99.05%)	15,935,102	0xebec795c [37]	Liquidation
0x0f5f5358 [38]	124.47 (99.99%)	15,953,996	0xba401cda [39]	Arbitrage
0x35ec8709 [40]	93.12 (99.99%)	15,995,535	ijcole.eth [41]	Arbitrage
0x4b6e0c33 [42]	74.62 (99.23%)	15,952,167	Lido [7]	Arbitrage
0xaebf69df [43]	71.26 (98.27%)	17,241,274	0xdafea492 [33]	Arbitrage
0x0938e0d7 [44]	64.01 (81.50%)	17,214,239	0x95222290 [45]	Arbitrage
0x6dce19ac [46]	56.78 (99.07%)	15,954,105	Lido [7]	Arbitrage
0x18ba0501 [47]	55.02 (99.47%)	16,227,778	0xcbfa8840 [48]	Sandwich

TABLE 4: Top 10 private transactions sorted by their profits to validators in our 15.5-month dataset. *Profits (ETH)* denotes the private transaction's profits and its percentage in the block's total private transaction profits.

5.1.3. Private Transactions. We examine the top 10 most profitable private transactions and find that all of them are related to MEV.

Top 10 private transactions. Table 4 displays the top 10 private transactions sorted by the profits. The greater the potential profits for attackers, the stronger the incentive for them to carry out their attacks. Notably, all the top 10 transactions belong to MEV, comprising 1 liquidation, 1 sandwich, and 8 arbitrage transactions. The highest-yielding private transaction [15] offers 541.90 *ETH*, equivalent to 1,282,005.34 USD on January 1, 2024.

MEV. MEV represents the maximum value that can be extracted from ordering transactions, mainly including liquidation, arbitrage, frontrun, backrun, and sandwich, as defined by Zeromev [49]. Our data analysis incorporates all these MEV types obtained from ZeroMEV. Liquidation allows a liquidator to repay a debt and take collateral, arbitrage profits from token price differences, and sandwich attacks trap a user's transaction to make a profit consisting of frontrun and backrun. In a frontrun, the attack transaction precedes the victim's, while in a backrun, the attack transaction follows the victim's. In the case of frontrun, backrun, and sandwich scenarios, we treat each instance as a single MEV case rather than considering all the transactions involved. Table 5 summarizes the profits from all the 7,620,084 MEV related private transactions, where liquidation has the most profits on average (0.225 *ETH*). It specifically highlights the percentage of MEV private transactions that meet the profit criteria set by our *prospective* reorganization attacks. We find that

MEV	#Private	Profits (*ETH*)	> average (0.01 *ETH*)	> 0.2 *ETH*
Liquidation	6,037	0.225	3,121 (51.70%)	483 (8.00%)
Arbitrage	1,433,287	0.019	261,371 (18.24%)	11,142 (0.78%)
Frontrun	1,914,625	0.007	220,522 (11.52%)	8,725 (0.46%)
Backrun	1,921,431	0.048	1,054,086 (54.86%)	84,842 (4.42%)
Sandwich	2,344,704	0.008	257,197 (10.97%)	3,788 (0.16%)

TABLE 5: The summary of MEV related private transactions in our 15.5-month dataset. *#Private* represents the count of private transactions. *average (0.01 ETH)* represents the average profits from all the private transactions in our dataset is 0.01 *ETH*.

that arbitrage (51.70%) and backrun (54.86%) have the most profitable private transactions, with the required profit set at the average (0.01 *ETH*), whereas sandwich transactions (10.97%) exhibit the fewest profitable instances. In total, 1,796,297 (23.57%) of the total 7,620,084 MEV related private transactions have profits more than average. Additionally, even when the required profit is increased to 0.2 ETH, arbitrage (8.00%) and backrun (4.42%) continue to lead in lucrative private transactions, while sandwich transactions (0.16%) remain the least profitable.

Finding 3: We analyze the profits of private transactions from a 15.5-month dataset, consisting of 30,062,233 private transactions and 2,500,979 blocks (with at least one private transaction). We find that 744,770 (29.78%) blocks having more profits than average (0.13 *ETH*) and 6,625,804 (22.04%) private transactions having more profits than average (0.01 *ETH*). 23.57% of such private transactions can provide enough profits to motivate our attacks, and all the top 10 lucrative private transactions are MEV related.

5.2. Attack Analysis

To confirm the feasibility of our attacks, we measure empirical datasets to find cases satisfying our attack requirements. We first introduce our datasets and then study the matched attack cases.

5.2.1. Datasets. We collect the necessary information for our 15.5-month dataset from September 15, 2022 (slot 4,700,013) to December 31, 2023 (slot 8,103,598), containing a total of 3,403,586 slots. To analyze the real-world attack cases, we collect the following information from reliable sources:

- **Forked slot.** We collect all forked slots, identifying 8,325 instances through information gathered from Beaconcha.in [50], and every forked slot represents a potential attack case. For example, the proposed block in slot 4,705,617 is forked.
- **Validator label.** To determine the identity of proposers, we extract the label of validator from QuickNode [51]. For instance, the proposer of slot 4,705,617 is associated with *Pool: lido*.
- **Votes.** We retrieve information on votes for specific slots from QuickNode. As an illustration, there are 2,831 votes for the forked block in slot 4,705,617.

5.2.2. Reorganization Attacks. We find potential attack cases based on our attack requirements (see §4).

Attack detection rules. We exclusively outline the attack criteria for *prospective* reorganization attacks, as *retrospective* attacks are seldom feasible in real-world scenarios due to the demanding requirement of at least 66.67% staking for attackers. In the context of *prospective* reorganization attacks, we identify attack scenarios according to the following rules:

R1: reorganized block. Considering three adjacent slots, which are slot $n+1$, slot $n+2$, and slot $n+3$. Block in slot $n+2$ is forked, whereas blocks in slot $n+1$ and slot $n+3$ are mined.

R2: block chain. The reorganized block proposed in slot $n+2$ should be attached to the block proposed in slot n, since it has no view of the attack block proposed in slot $n+1$.

R3: proposer identify. The proposer of slot $n+2$ belongs to the mining pool H, while the proposers of slot n+1 and slot n+3 belong to A. H is required to be different from A.

R4: no votes for the reorganized block. There should be no votes for the reorganized block proposed in the forked slot $n+2$, as the attack has already succeeded at the start of the forked slot. Considering the abnormal votes, it is required that the votes in this slot should be no more V.

For the *Prospective - proposer boost* reorganization attacks, we find the attack cases based on four rules as well. The first three rules mirror the rules outlined above, and the fourth rule is as follows:

R4: votes for the reorganized block. Votes are expected for the reorganized block proposed from the forked slot $n+2$, as the attack has not succeeded, in contrast to the *prospective* reorganization attack without the proposer boost mechanism.

Attack cases. Table 6 showcases the measurement results, revealing a total of 124 potential matched attack cases for *prospective* attacks, whether the proposer boost mechanism is employed or not. In real-world voting scenarios, we place constraints on abnormal votes, which occur when honest validators support the attack block or attackers endorse the honest block. We set a limit on such instances, ensuring they do not exceed V per slot. The higher the value of V, the more instances of *prospective* attacks without the proposer boost mechanism and the fewer instances of *prospective* attacks with the proposer boost mechanism. In our analysis, we set several possible values of V and measure the attack cases of these two types of *prospective* attacks. We can tell that no matter what V is, there are always more *prospective* attacks with proposer boost mechanism. We suspect most validators adopt the proposer boost mechanism. Furthermore, we examine the gains associated with private transactions in the 124 potential attack cases. Specifically, we calculate the profits derived from the block proposed in slot $n+2$, which appropriates gains from private transactions. Notably, the average profits within these blocks (0.23 *ETH*) significantly surpass the overall average profits across all blocks (0.13 *ETH*).

#votes (V)	0	10	50	100	500	1,000
#Cases of Prospective	3	9	10	11	12	13
#Cases of Prospective - Proposer Boost	121	115	114	113	112	111

TABLE 6: Summary of potential real-world attack cases matching our reorganization attacks in the 15.5-month PoS dataset under varying V, which represents the maximum number of allowed abnormal votes per slot.

Finding 4: In our 15.5-month dataset, we identify 124 potential instances of *prospective* reorganization attacks. Notably, instances of *prospective* attacks utilizing the proposer boost mechanism outnumber those without it. Furthermore, the average profits observed in the attack blocks from these potential instances amount to 0.23 *ETH*, significantly surpassing the average profits derived from all blocks, which stand at 0.13 *ETH*.

6. Related Works

We first provide a thorough overview of the most pertinent studies concerning consensus-level attacks akin to our reorganization attacks in §6.1. Additionally, we delve into other related works, such as the measurement of private transactions in §6.2.

6.1. Prior works on consensus level attacks

In PoS Ethereum, the most intuitive consensus level attack is the 33% attack [52] where attackers control more than 1/3 validators to prevent the blockchain from reaching finality; however, it does not target any specific transaction for gaining profits. Similarly, other works [27], [14] explore malicious reorganization attacks in PoS Ethereum without targeting any transactions. Notably, *Neuder et al.* [27] present the 30% attack, where attackers strategically withhold their proposed blocks until honest validators propose the next block, and thus, attackers successfully fork the blockchain. Attack will succeed if the attack chain gains more votes than the honest chain, which requires at least 30% staking. Furthermore, *Schwarz-Schilling et al.* [14] propose a refined ex ante reorganization attack in PoS Ethereum, similar to ours, resulting in a high attack success rate with only 0.09% controlled staking. In PoW Ethereum, works by *Carlsten et al.*[53] and *Gong et al.*[54] delve into undercutting attacks. These attacks occur when transaction profits surpass block rewards, and miners intentionally fork the blockchain to leave lucrative transactions unclaimed, enticing other miners to create blocks on the new chain. In our paper, we propose *retrospective* and *prospective* reorganization attacks in PoS Ethereum, building upon existing ex post and ex ante reorganization attack strategies [14]. However, we focus on leveraging these attacks to leak private transactions for profits.

We compare our work with existing works in Table 7. While existing works assess risks of consensus protocols [52], [27], [14] or propose attacks targeting at public transactions [53], [54], our work focuses on leaking private transactions and making profits via reorganization attacks.

Work	Protocol	Attack	Transaction	Profit	Staking/Hashrate
[52]	PoS	No Finalization	N/A	No	33.34%
[27]	PoS	Reorganization	N/A	No	30.00%
[14]	PoS	Reorganization	N/A	No	0.09%
[53][54]	PoW	Undercutting	Public	Yes	Low
This Paper	PoS	*Retrospective* (Appendix B)	Private	Yes	66.67% (100.00%)
	PoS	*Prospective* (§4.1)	Private	Yes	1% (21.15%)
	PoS	*Prospective - Proposer Boost* (§4.2)	Private	Yes	Table 3

TABLE 7: Summary of comparison with related works on consensus level attacks. *Staking/Hashrate* represents the required staking percentage (p) in PoS Ethereum or hashrate percentage in PoW Ethereum controlled by attackers, along with the success rate of launching at least one attack per day (P). The staking percentage required by *Prospective - Proposer Boost* reorganization attack depends on the proposer boost score (PB), as shown in Table 3 (e.g., when PB = 90%, p = 3.34%, P = 92.49%).

1) Launching a retrospective reorganization attack is unfeasible as it necessitates a 66.67% hashrate in PoS Ethereum, a condition that contrasts with the feasibility of such attacks in Ethereum. That is, the goal of targeting posted private transactions for profits remains unattainable through *retrospective* reorganization attacks in PoS Ethereum.

2) Our proposed prospective reorganization attacks entail slightly higher staking/hashrate requirements compared to certain existing works, primarily due to our distinct goal of targeting private transactions for profits. In the case of malicious reorganization attacks in PoS Ethereum as discussed in [14], the success merely necessitates 0.09% staking. In contrast, our *prospective* reorganization attack demands 1% staking, owing to its specific requirement of mining an attack block to include leaked private transactions and subsequently exploit their profits (see §4.1). The attack presented in [14], on the other hand, does not require the creation of an attack block.

3) The inherent nature of private transactions significantly impacts both attack strategies and analysis. Private transactions are exclusively visible to specific validators. This uniqueness sets our proposed attacks apart from those targeting public transactions. For instance, certain attacks that are effective on public transactions, such as undercutting attacks [53], [54], do not apply to private transactions. Unlike public transactions, private transactions cannot be left to entice honest validators to join the attack fork, as they remain visible solely to specific validators.

4) Our work is different from selfish mining attacks. Selfish mining attacks [55], [56], [57], [58], [59] share similarities with our *prospective* reorganization attacks (e.g., both belong to reorganization/block withholding attacks); however, they aim at maximizing mining rewards, instead of focusing on private transactions, and they are hardly applicable in PoS Ethereum due to the mining mechanism, which randomly assigns validators for mining new blocks.

6.2. Other related works

Measuring Private Transactions. Previous works [8], [9], [10], [11], [3], [12] have studied private transactions in Ethereum, with a focus on MEV and blockchain extractable value (BEV) in private transactions based on PoW Ethereum. *Qin et al.* [8] studied the participation of mining pools in private transactions and measured the percentage and value of private transactions in each BEV category. *Capponi et al.* [9] proposed a game theoretic analysis with the profits from private transactions and obtained empirical evidence. *Piet et al.* [10] measure private transactions in PoW to study MEV and found that 91.5% of MEV transactions are private transactions. *Weintraub et al.* [11] mainly studied private transactions providers (Flashbots) and MEV extraction in PoW. They drew the similar conclusion that most MEV extraction comes from private service providers. *Lyu et al.* [3] collected a one-year private transactions dataset in PoW and performed an empirical study on private transactions in terms of their characteristics, economics impacts, and security impacts. *Yang et al.* [12] mainly use private transactions to measure 28 MEV auction platforms in PoS Ethereum, in terms of their market shares, the relationship of internal components, and the related security guarantees. Different from these works, our work aims to shed lights on leaking private transactions for profits in both PoW and PoS Ethereum.

Quantifying MEV. Many works [60], [8], [61], [62], [9], [10], [11], [3], [12] have quantified MEV extraction from transactions in Ethereum. In particular, *Daian et al.* [60] were the first to propose and quantify MEV. *Qin et al.* [8] primarily focused on measuring BEV, which includes MEV. *Torres et al.* [61] identified three types of frontrunning, including displacement, insertion, and suppression. *Wang et al.* [62] proposed a theoretical framework to study cyclic arbitrage and detect 292,606 cyclic arbitrages over eleven months. *Capponi et al.* [9] examined the economic impact of MEV. *Piet et al.* [10] analyzed MEV usage and profit redistribution. They found that miners take the most profits from MEV extraction. *Weintraub et al.* [11] found that MEV extraction mainly comes from Flashbots, with miners gaining more profits than MEV searchers. *Lyu et al.* [3] studied the MEV extraction and profit distribution in private transactions. *Yang et al.* [12] systematized the knowledge of the theory and practice of 28 MEV auction platforms. These works have mainly focused on identifying and quantifying MEV in PoW transactions or MEV auction platforms in PoS Ethereum, while our work examines MEV in private transactions, to analyze potential profits attackers can earn in our reorganization attacks.

Measuring Proof-of-Stake (PoS) Ethereum. Several papers [63], [64], [65], [66], [67] have examined the blockchain PoS or PoS-similar consensus mechanisms. *Kapengut et al.* [63] studied the impact of PoS on the Ethereum network and other blockchains (e.g., Polygon [68]). *Du* et al. [64] analyzed various consensus algorithms, including their basic characteristics and application scenarios. *Lepore et al.* [65] measured the performance of PoS, and compared it with other consensus mechanisms, based on throughput and scalability. *Saad* [66] studied the key parameters used in PoS and Delegated Proof of Stake (DPos). *Cao et al.* [67] investigated three consensus mechanisms, including PoW, PoS, and Direct Acyclic Graph

(DAG), to measure their performance. While these works mainly focus on measuring the network performance in PoS, our study centers on the reorganization attacks for leaking private transactions.

7. Discussion

We discuss the limitations of our study that can be addressed in future work, private transaction datasets, and attack mitigations.

Limitations. First, our attacks only consider 1-block reorganization. This is because the probability of multiple-reorganization or multiple attack block attacks is too insignificant to consider. Second, there are other similar attacks where malicious validators attempt to fork the blockchain for maximizing profits, such as the time bandit attack [69]. While these attacks share similarities with reorganization attacks, they fall beyond the scope of our current study. Third, we only perform detailed analysis on MEV private transactions, whose profits directly incentivize the reorganization attacks in our model. Furthermore, constrained by practical limitations, our analysis on liquidation only focuses on two popular Decentralized Finance (DeFi) - Aave and Compound. The intricate nature of MEV across various platforms renders a thorough measurement unattainable. Analyzing data from these platforms offers insights into MEV, though limitations exist due to data constraints. Fourth, although our paper outlines the rules for detecting suspicious attack cases, achieving 100% accuracy poses a challenge due to network delays. For example, in cases where a block is proposed with delays, discerning whether the block proposer's intention is benign or malicious becomes difficult, thus complicating the determination of whether such actions are part of an attack.

Private transaction datasets. Private transactions are designed to bypass public networks, sent directly to block creators to evade monitoring. This practice is endorsed by major blockchain companies like Alchemy [70], Blocknative [32], and Zeromev [49]. Our study utilizes Blocknative's datasets for private transaction labels. They monitor transactions, classifying those not observed by any node yet later mined as private. Similarly, Sen et al. [12] use a comparable approach. However, we lack resources for long-term monitoring with multiple nodes, hence opting for Blocknative's dataset, a widely utilized resource [71], [72]. To ensure high accuracy, Blocknative operates multiple global nodes. Additionally, we have manually validated a subset of 100 randomly sampled private transactions. However, rare misclassification can happen due to network latency.

Attack mitigations. Formulating a comprehensive protocol to mitigate attacks poses significant challenges, often necessitating dedicated research and exploration. While solutions like the proposed Proper Boost Score (PBS) from the Ethereum community aim to mitigate reorganization attacks, their effectiveness in completely eliminating such threats remains uncertain. However, ongoing initiatives within Ethereum, such as the pursuit of single slot finality, hold promise in deterring reorganization attacks. We also propose additional strategies to fortify against reorganization attacks.

1) Embracing single slot finality. Achieving finality within a single slot interval marks a significant milestone in enhancing blockchain security. Finality not only instills an elevated level of trust in the validity of blocks but also fortifies them against reorganizations once attained. Currently, Ethereum's finalization process spans approximately 15 minutes, equivalent to processing 75 blocks at a rate of 12 seconds per block. However, by streamlining this process to synchronize with 1 block, the potential to markedly reduce susceptibility to reorganization attacks emerges [73]. By condensing the finalization timeline to match the slot time, Ethereum can fortify its defenses against our proposed reorganization attacks, augmenting its overall security posture.

2) Adjusting incentivization mechanisms. The current incentivization mechanism in Ethereum, particularly concerning the rewarding of validators for attestations, presents an avenue for potential mitigation against our proposed attacks. One potential adjustment involves restructuring the reward system to eliminate partial rewards for flawed attestations. This adjustment would significantly raise the profitability threshold for executing reorganization attacks, making the financial requirements for such attacks considerably more challenging. However, this adjustment needs comprehensive discussions and considerations before implementation.

8. Conclusion

We have presented the *first* feasibility study on actively leaking private transactions for profits, by introducing two types of reorganization attacks: *retrospective* and *prospective* to expose private transactions and appropriate their profits. We detail the attack models and strategies, taking into account staking requirements and the minimum profit necessary to incentivize these attacks. By analyzing real-world datasets, we find that private transactions related to MEV are the primary sources of profits. Moreover, we explore potential attack cases that match our attack patterns in real-world data. Our study underscores the financial risks linked to the leakage of private transactions and aims to inspire further research on this topic, ultimately driving the development of more effective countermeasures.

Acknowledgment

We thank the anonymous reviewers for their insightful suggestions and comments. This research was partly supported by a gift from Forta Network. Yinqian Zhang was also supported in part by the Shenzhen Science and Technology Program under Grants JSGG202208310956030.

References

[1] Alchemy, "How to send private transactions on ethereum." https://www.alchemy.com/overviews/ethereum-private-transactions, 2022.

[2] Taichi-Network, "Taichi-network." https://github.com/Taichi-Network, 2022.

[3] X. Lyu, M. Zhang, X. Zhang, J. Niu, Y. Zhang, and Z. Lin, "An empirical study on ethereum private transactions and the security implications," *arXiv preprint arXiv:2208.02858*, 2022.

[4] Flashbot, "Flashbots docs." https://docs.flashbots.net/flashbots-auction/overview/, 2022.

[5] Etherscan, "Transaction details." https://etherscan.io/tx/0xc2969993a652dec6a3ba52b3437bd9cac9bdd5153d2962b9e843ebcaaa580139, 2022.

[6] Etherscan, "Transaction details." https://etherscan.io/tx/0x53a6f3f1d483d3a6c6132f8833fa8065cf7e2bae97426bdb20c9a0912dec1359, 2022.

[7] Etherscan, "Lido: Execution layer rewards vault." https://etherscan.io/address/0x388c818ca8b9251b393131c08a736a67ccb19297, 2023.

[8] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?," *arXiv preprint arXiv:2101.05511*, 2021.

[9] A. Capponi, R. Jia, and Y. Wang, "The evolution of blockchain: from lit to dark," *arXiv preprint arXiv:2202.05779*, 2022.

[10] J. Piet, J. Fairoze, and N. Weaver, "Extracting godl [sic] from the salt mines: Ethereum miners extracting value," *arXiv preprint arXiv:2203.15930*, 2022.

[11] B. Weintraub, C. F. Torres, C. Nita-Rotaru, and R. State, "A flash (bot) in the pan: Measuring maximal extractable value in private pools," *arXiv preprint arXiv:2206.04185*, 2022.

[12] S. Yang, F. Zhang, K. Huang, X. Chen, Y. Yang, and F. Zhu, "Sok: Mev countermeasures: Theory and practice," *arXiv preprint arXiv:2212.05111*, 2022.

[13] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, p. 21260, 2008.

[14] C. Schwarz-Schilling, J. Neu, B. Monnot, A. Asgaonkar, E. N. Tas, and D. Tse, "Three attacks on proof-of-stake ethereum," 2021.

[15] Etherscan, "Transaction details." https://etherscan.io/tx/0xa954b21c4ca74672c116ba517fce923b295c10d3b79a3fbb947a1eab7e5ee869, 2023.

[16] V. Buterin, "Ethereum white paper: A next generation smart contract & decentralized application platform," 2013.

[17] E. I. Proposals, "Eip-1559: Fee market change for eth 1.0 chain." https://eips.ethereum.org/EIPS/eip-1559, 2022.

[18] Ethereum, "Proof-of-stake (pos) | ethereum.org." https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/, 2022.

[19] B. Edgington, "Randomness." https://eth2book.info/bellatrix/part2/building_blocks/randomness/\#fn-4, 2024.

[20] Github, "how does pos ethereum prevent bribery for late block proposal? #111." https://github.com/flashbots/mev-boost/issues/111, 2022.

[21] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, and Y. X. Zhang, "Combining ghost and casper," *arXiv preprint arXiv:2003.03052*, 2020.

[22] Ethereum, "Proposer lmd score boosting." https://github.com/ethereum/consensus-specs/pull/2730, 2021.

[23] CoinTelegraph, "Bnb smart chain hit with copycat vyper attack, $73k exploited." https://cointelegraph.com/news/vyper-copycat-exploit-on-bsc-bnb-smart-chain-curve, 2023.

[24] J. Bonneau, "Why buy when you can rent? bribery attacks on bitcoin-style consensus," in *Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers 20*, pp. 19–26, Springer, 2016.

[25] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*, pp. 264–279, Springer, 2017.

[26] P. McCorry, A. Hicks, and S. Meiklejohn, "Smart contracts for bribing miners," in *Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers 22*, pp. 3–18, Springer, 2019.

[27] M. Neuder, D. J. Moroz, R. Rao, and D. C. Parkes, "Low-cost attacks on ethereum 2.0 by sub-1/3 stakeholders," 2021.

[28] B. Edgington, "Randomness." https://eth2book.info/altair/part2/building_blocks/randomness/, 2024.

[29] B. Edgington, "Part 3: Annotated specification fork choice." https://eth2book.info/capella/part3/forkchoice/phase0/, 2024.

[30] G. Ethereum, "Official go implementation of the ethereum protocol." https://github.com/ethereum/go-ethereum, 2022.

[31] EthereumETL, "Ethereum etl." https://github.com/blockchain-etl/ethereum-etl, 2023.

[32] Blocknative, "Blocknative relay." https://www.blocknative.com/, 2024.

[33] Etherscan, "Flashbots: Builder." https://etherscan.io/address/0xdafea492d9c6733ae3d56b7ed1adb60692c98bc5, 2023.

[34] Etherscan, "Transaction details." https://etherscan.io/tx/0x0bff9cfabf3e532cd30f94cc2cb17a491f3209e9be9140834025a2cc6d7f6b61, 2022.

[35] Etherscan, "Address." https://etherscan.io/address/0xee5f5c53ce2159fc6dd4b0571e86a4a390d04846, 2023.

[36] Etherscan, "Transaction details." https://etherscan.io/tx/0xe336debd021beac7bcd1bdb3c52f76bb124a67902ed34e932293aae7ee610cc3, 2022.

[37] Etherscan, "Address." https://etherscan.io/address/0xeBec795c9c8bBD61FFc14A6662944748F299cAcf, 2023.

[38] Etherscan, "Transaction details." https://etherscan.io/tx/0x0f5f53589a6143c9aafbdc22e7ab9e33a681b84df181b455d2294ccbf4a44eb9, 2022.

[39] Etherscan, "Address." https://etherscan.io/address/0xBA401CdaC1A3b6AEeDe21c9C4a483be6C29F88C5, 2023.

[40] Etherscan, "Transaction details." https://etherscan.io/tx/0x35ec870912db36ec935aeeb9a3dacdc90158c288428290cf43a77a9ffe788279, 2022.

[41] Etherscan, "Address." https://etherscan.io/address/0xc88386be9463EcB08b246386D14AAd2F48159A1d, 2023.

[42] Etherscan, "Transaction details." https://etherscan.io/tx/0x4b6e0c33b24b3cfddf29f2722883ad1932f23b24875bab34629ce5379614f36b, 2022.

[43] Etherscan, "Transaction details." https://etherscan.io/tx/0xaebf69dfc4e364652096575bcb56d25be64c149a2faa7c1d939e57ac6b33918a, 2022.

[44] Etherscan, "Transaction details." https://etherscan.io/tx/0x0938e0d7f754637ed588a769889ae30d70ac8190d6f4c6c707529885333d00e0, 2022.

[45] Etherscan, "beaverbuild." https://etherscan.io/address/0x95222290dd7278aa3ddd389cc1e1d165cc4bafe5, 2023.

[46] Etherscan, "Transaction details." https://etherscan.io/tx/0x6dce19ac4fe6a2b38268e8933d6400e17d43f72a50695429e36e325c3d487e2a, 2022.

[47] Etherscan, "Transaction details." https://etherscan.io/tx/0x18ba050195851c8018fd33bd6fc56d2277b3727e60b8f1e4757ad33eaaa84847, 2022.

[48] Etherscan, "Address." https://etherscan.io/address/0xcBfa884044546d5569E2abFf3fB429301b61562A, 2023.

[49] Pmcgoohan, "zeromev." https://www.zeromev.org/, 2023.

[50] Beaconcha.in, "Open source ethereum explorer." https://beaconcha.in, 2024.

[51] QuickNode, "Eth v1 beacon states State Id committees." https://www.quicknode.com/docs/ethereum/eth-v1-beacon-states-\%7Bstate_id\%7D-committees, 2023.

[52] Ethereum, "Ethereum proof-of-stake attack and defense." https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/attack-and-defense/, 2024.

[53] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 154–167, 2016.

[54] T. Gong, M. Minaei, W. Sun, and A. Kate, "Towards overcoming the undercutting problem," in *International Conference on Financial Cryptography and Data Security*, pp. 444–463, Springer, 2022.

[55] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

[56] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 195–209, 2017.

[57] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 305–320, IEEE, 2016.

[58] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016, Revised Selected Papers 20*, pp. 515–532, Springer, 2017.

[59] C. Feng and J. Niu, "Selfish mining in ethereum," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1306–1316, IEEE, 2019.

[60] 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 *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 910–927, IEEE, 2020.

[61] C. F. Torres, R. Camino, and R. State, "Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain," *arXiv preprint arXiv:2102.03347*, 2021.

[62] Y. Wang, Y. Chen, H. Wu, L. Zhou, S. Deng, and R. Wattenhofer, "Cyclic arbitrage in decentralized exchanges," *Available at SSRN 3834535*, 2022.

[63] E. Kapengut and B. Mizrach, "An event study of the ethereum transition to proof-of-stake," *arXiv preprint arXiv:2210.13655*, 2022.

[64] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 2567–2572, IEEE, 2017.

[65] C. Lepore, M. Ceria, A. Visconti, U. P. Rao, K. A. Shah, and L. Zanolini, "A survey on blockchain consensus with a performance comparison of pow, pos and pure pos," *Mathematics*, vol. 8, no. 10, p. 1782, 2020.

[66] S. M. S. Saad and R. Z. R. M. Radzi, "Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos)," *International Journal of Innovative Computing*, vol. 10, no. 2, 2020.

[67] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[68] J. Kanani, S. Nailwal, and A. Arjun, "Matic whitepaper," *Polygon, Bengaluru, India, Tech. Rep., Sep*, 2021.

[69] A. Judmayer, N. Stifter, P. Schindler, and E. Weippl, "Estimating (miner) extractable value is hard, let's go shopping!," *Cryptology ePrint Archive*, 2021.

[70] Alchemy, "How to send a private transaction on ethereum." https://docs.alchemy.com/docs/how-to-send-a-private-transaction-on-ethereum, 2024.

[71] A. Adams, B. Y. Chan, S. Markovich, and X. Wan, "The costs of swapping on the uniswap protocol," *arXiv preprint arXiv:2309.13648*, 2023.

[72] L. Heimbach, L. Kiffer, C. Ferreira Torres, and R. Wattenhofer, "Ethereum's proposer-builder separation: Promises and realities," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, pp. 406–420, 2023.

[73] Ethereum, "Single slot finality." https://ethereum.org/en/roadmap/single-slot-finality/, 2023.

[74] Beaconcha.in, "Block 16343639." https://beaconcha.in/block/16343639\#attestations, 2023.

Appendix

1. Private Transaction Leakage

We notice that the private transactions have already been leaked to some extent; therefore, validators/miners can easily obtain the existing leaked private transactions for profits. To quantify the leakage, we deployed two modified Ethereum nodes (Geth [30] in full mode) on different continents and monitored the transactions coming to the local mempool where transactions reside and wait to be mined.

Leakage. In PoS Ethereum, there is a 0.51% private transaction leakage and all the leakages are caused by block reorganization. We monitor local mempools for 14 days, from December 17, 2022, to December 30, 2022, collecting a total of 13,262,153 transactions from Node 1 and 12,255,420 transactions from Node 2 during this period. We then compare these transactions to the total 367,526 private transactions from Blocknative, and present our findings in Table 8. Our analysis reveals that some private transactions are not guaranteed as private, with an average leakage of 0.51% over the 14-day period. The highest number of leaked private transactions were found on day 7 (December 23, 2022) at 201 (0.85%) and the least on day 2 (December 18, 2022) at 77 (0.24%). We suspect that these transactions were leaked due to block reorganization, which occurs when a block loses competition with other blocks and is no longer part of the canonical chain.

Day	Node 1 View	Node 2 View	Two Nodes Combined	Labeled by Blocknative	Found in Local	Found in Reorganized
1	1,021,957	826,323	1,049,730	25,333	89 (0.35%)	89 (100%)
2	921,606	842,702	9,681,157	32,072	77 (0.24%)	77 (100%)
3	1,033,281	862,601	1,056,566	29,064	142 (0.49%)	142 (100%)
4	980,193	873,010	1,042,373	27,444	106 (0.39%)	106 (100%)
5	983,347	958,026	1,007,291	28,574	127 (0.44%)	127 (100%)
6	958,470	872,117	983,470	27,521	150 (0.51%)	150 (100%)
7	1,024,765	949,059	1,065,982	23,502	201 (0.85%)	201 (100%)
8	872,413	882,524	927,602	22,477	175 (0.78%)	175 (100%)
9	827,962	801,058	891,007	27,243	194 (0.71%)	194 (100%)
10	908,785	871,154	916,261	26,323	152 (0.58%)	152 (100%)
11	982,947	892,721	1,030,714	25,670	116 (0.45%)	116 (100%)
12	912,749	900,004	937,322	24,973	97 (0.38%)	97 (100%)
13	917,685	858,006	927,459	22,214	109 (0.49%)	109 (100%)
14	915,919	866,115	958,287	25,116	126 (0.50%)	126 (100%)

TABLE 8: Summary of private transactions observed by our two nodes and comparison with Blocknative [32] in 14 days.

To confirm this hypothesis, we record the transaction hash, the number of the block where it was mined, and the timestamp of the block for every transaction that came to our local mempools. We discover that if a private transaction is found mined in block B1 and later in block B2, then block B1 becomes a reorganized block and this private transaction is leaked, as it will be broadcast in the P2P network after the reorganization. Our analysis indicate that all the leaked private transactions are found in reorganized blocks.

2. Retrospective Reorganization Attack

Attack model. Fig. 5 illustrates the *retrospective* reorganization attack, given two adjacent slots within the same epoch. In PoS Ethereum, each slot has an assigned proposer for proposing one block. Every proposed block might contain a few private transactions whose profits will be obtained by the block proposer mining this block. We assume block m+1 proposed in slot $n+1$ has the private transaction PTx-99 with 99 profits and the private transaction PTx-8 with 8 profits, whereas block m+2 proposed in slot $n+2$ has the private transaction PTx-12 with 12 profits. Specifically, the profits stemming from a private transaction encompass two crucial elements: the direct payment made from the user to the proposer and the tips received by the proposer from the transaction fee after deducting the burnt fee. We further assume that honest validators are selected to propose a block in slot $n+1$, while attackers are designated to propose a block in slot $n+2$. To be specific, attackers here represent a group of malicious validators. Without attacks, the blockchain will remain as the expected chain, where typically one block is proposed at one slot and private transactions will be mined into the related block. However, attackers may launch a *retrospective* reorganization attack driven by the huge profits of private transactions (e.g., PTx-99) from the block m+1, after they observe this block is proposed by honest validators.

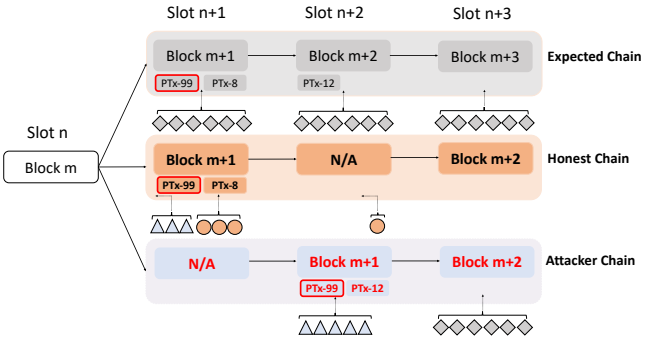

Figure 5: The *retrospective* reorganization attack incentivized by private transactions, ending with leaking private transactions PTx-99 and PTx-8 and stealing the profits of PTx-99 from the honest block proposer. *PTx-i* represents the private transaction with profits i. ◇ represents a vote by either attacker or honest validators; ○ represents a vote in the block of the honest chain controlled by honest validators; △ represents a vote in the block of the attacker chain controlled by attackers. All the attack validators in slot $n+1$ vote for block m in slot n and all the honest validators in slot $n+2$ vote for block m+1 in slot $n+1$.

Attack strategy. The attackers intentionally construct a fork with the following two steps. To be specific, the honest block has already been mined in the blockchain. Since the blockchain is public to everyone, the attacker can get to know every transaction details, such as the profits of private transactions and the validator attestation distribution. Therefore, attackers can know whether this attack will be successful or not since they also know the distribution of committee validators in advance. If the attack can bring profits to the attackers, it is very likely they will launch the attack.

Step 1: target honest block in slot $n+1$. In slot $n+1$, attackers compel the controlled M_1 validators to vote for

block m in slot n, while the remaining $N - M_1$ honest validators vote for the honest block m+1 in slot $n + 1$.

Step 2: steal profits in slot $n + 2$. In slot $n+2$, the attackers intentionally construct a forked chain by appending the block m+1 proposed in slot $n+2$ to block m in slot n, instead of the block in slot $n + 1$ proposed by honest validators. To succeed in the attack, the attacker chain must receive more votes from attestation validators than the honest chain. Assuming attackers control M_2 validators in slot $n + 2$ and all the attack validators vote for the attack block m+1, the remaining $N - M_2$ honest validators vote for the honest chain. Concurrently, the attackers pilfer profits from private transactions in the honest block, presuming the success of the attack.

Attack example. As shown in Fig. 5, let's assume each slot comprises N = 6 attestation validators, with attackers controlling M_1 = 3 in slot $n + 1$, voting for the block m. Ideally, the remaining 3 honest validators will vote for the honest chain in slot $n + 1$. Furthermore, let's assume attackers control M_2 = 5 in slot $n + 2$, with the remaining 1 honest validator voting for the attacker chain. At this point, the attacker chain accumulates more votes (5 from slot $n + 2$) than the honest chain (3 from slot $n + 1$ and 1 from slot $n + 2$), leading to a successful attack. Consequently, all 6 attestation validators in slot $n + 3$ will vote for the attack chain, regardless of whether they are attack or honest validators. As the attack succeeds, PTx-8 and PTx-99 are leaked into the public P2P network, rendering them no longer private. Consequently, attackers can pick any private transactions from the set of PTx-99, PTx-12, and PTx-8 for mining. Intuitively, attackers will mine as many as private transactions they can to maximize their profits. However, assume there are conflicts between PTx-12 and PTx-8 (e.g., two MEV searchers targeting at the same MEV opportunity), PTx-12 will be mined and PTx-8 will be discarded. Therefore, while both PTx-99 and PTx-8 are leaked, only the profits of PTx-99 are stolen by attackers.

Attack staking requirement and success rate. In accordance with the attack strategy, the staking requirement for the attack is as follows:

$$M_2 > (N - M_1) + (N - M_2). \tag{13}$$

Considering $M_1 \approx M_2 \approx p \times N$, the staking requirement translates to:

$$P_v : p > 2/3. \tag{14}$$

The necessary staking ($> 2/3$) for our proposed retrospective reorganization attack is impractical to achieve, based on the observation that the stake typically does not exceed $1/3$ in current PoS Ethereum [52]. Moreover, our attack requires the honest validator is the proposer of slot $n+1$ and the attacker is the proposer of slot $n+2$; therefore, the probability P_p of satisfying the proposer requirement is as follows:

$$P_p = p(1 - p). \tag{15}$$

Therefore, the success rate (P_{one}) of one attack given two adjacent slots under p should be the probability of satisfying both vote and proposer requirements, as follows:

$$P_{one} = P_v \times P_p. \tag{16}$$

Additionally, we assume that C denotes the number of attacks an attacker can initiate per day. With 7,200 slots and 225 epochs occurring each day, where each epoch comprises 32 slots, and considering that one attack necessitates two adjacent slots in one epoch, the calculation for C becomes 31 * 225 = 6,975. Therefore, the success rate (P) of launching at least one successful attack per day is as follows:

$$P = 1 - (1 - P_{one})^C. \tag{17}$$

Attack profits requirement. Suppose the attestation reward per validator is R_V, and the reward for mining a block proposed in slot i is R_{B_i}, which primarily consists of the block proposal reward and profits from all the mined transactions in that block, such as execution fees. Then we assume the total reward for attackers without launching the attack is R_{NA} shown as follows:

$$R_{NA} = (M_1 + M_2) \times R_V + R_{B_{n+2}}. \tag{18}$$

The total reward R_{AS} for the attacker successfully launching an attack is as follows (R_{ptx} represents the profits of private transactions from honest block stolen by attackers):

$$R_{AS} = (M_1 + M_2) \times R_V + R'_{B_{n+2}} + R_{ptx}. \tag{19}$$

Here $R'_{B_i} \approx R_{B_i}$ (i equals n+2) since the block proposed in the non-attack situation is almost the same as the block proposed in the attack, except that the attack block may adjust a few private transactions to maximize their profits (e.g., adding private transaction PTx-99), which is considered in R_{ptx}. We do not focus on the subtle changes caused by such adjustments. Therefore, we assume that the profits from blocks mined by attackers do not differ significantly from those blocks without attacks, except for the profits from target private transactions.

As R_{ptx} consistently exceeds or equals 0, indicating that R_{AS} is perpetually greater than or equal to R_{NA} at any given moment, a *prospective* reorganization attack doesn't hinge on any private transaction profits. However, there might be private transactions in the original attack blocks that could yield profits to attackers even without executing attacks. We suspect that despite meeting the attack staking requirement, the attack might not be executed due to the low profits from potentially leaked private transactions. *Only when the profits from potentially leaked private transactions surpass a threshold such as the average profits of all private transactions, will the attacks be launched (see §5.1.3 for real-world data analysis). Nevertheless, the impracticality of such attacks arises due to the exceptionally high staking requirements.*

3. Attack Simulation

Setup. To evaluate the success rate (P) of our proposed *retrospective* and *prospective* reorganization attacks, we perform extensive simulation experiments using the algorithm depicted in Alg. 1. This algorithm calculates the simulated success rate (P_{sim}), which represents the probability of launching at least one successful attack per day, considering different values of p (the staking percentage of the attacker) and t (representing the two types of reorganization attacks: *retrospective* or *prospective*).

Algorithm 1 Attack Simulation

Input: $n = 10,000, m = 225, q = 32, N = 28,224, p, t$
Output: P_{sim}

```
AttackTimes ← 0
for round ← 1 to n do
    for epoch ← 1 to m do
        votes ← GetOneEpoch(N, p)
        for slot ← 1 to q do
            if IsAttack(votes, slot, t) then
                AttackTimes ← AttackTimes + 1
                Exit both epoch and slot loops
            end if
        end for
    end for
end for
return AttackTimes/n
```

We configure the number of rounds (n) to be 10,000, and within each round, we simulate the attacks on a daily basis. The simulation is performed over 10,000 rounds, representing 7,200 slots per round, as each slot costs 12 seconds, and there are 225 epochs (m) per round, with each epoch containing 32 slots (q). For each epoch, we randomly assign the proposer and committee members for voting in each slot, based on the given committee member size (N equals 28,224 on January 1, 2024 [74]) and the percentage of staking controlled by attackers (p) via the function $GetOneEpoch(N, p)$. In each slot, we checked if the attackers can successfully launch an attack. For instance, if the function $IsAttack(votes, slot, t)$ returns true, the variable *AttackTimes* is incremented by 1 to represent the number of successful attacks launched per day. Specifically, $IsAttack(votes, slot, retrospective)$ returns true under the conditions: 1) the current slot's proposer is an attacker while the last slot's proposer is an honest validator, and 2) the votes for the attack chain exceed those of the honest chain. The algorithm concludes by computing the success rate (P) of attacks by dividing the total number of successful attack instances (*AttackTimes*) by the number of rounds (n).

Prospective Reorganization Attack. Fig. 6 presents the success rate derived from theoretical analysis, Equation (5) (denoted as P), and simulations (P_{sim}) across a range of staking percentages (p) within the interval [0.01, 0.05]. As expected, higher values of p correspond to higher success rates (P). Similar to *retrospective* reorganization attack, the simulation results align with the theoretical computation. Furthermore, launching a *prospective* reorganization attack is relatively straightforward. For instance, when p is 0.05, the success rate approaches almost 99.67%, as confirmed by both theoretical computation and simulation.

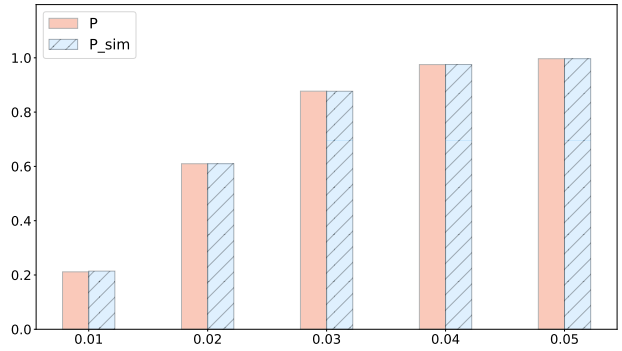

Figure 6: The success rate (P from Equation (5) and P_{sim} from simulation) of launching at least one successful *prospective* reorganization attack per day in PoS Ethereum, under different staking percentage (p) within [0.01, 0.05].

Prospective - Proposer Boost. The success rate resulting from theoretical analysis for the reorganization attack with the proposer boost mechanism is consistent with the attack without the proposer boost mechanism, both obtained from Equation (5) and denoted as P. As demonstrated in Table 3, the value of P is influenced by the proposer boost score (PB). When PB is less than or equal to 90%, P remains approximately 1, aligning with our simulation outcomes derived from the algorithm outlined in Alg. 1.

Finding 5: The extensive simulation experiments validate the feasibility of executing the attacks with the anticipated success rate. For instance, when the attackers' controlled staking is 0.05, both theoretical analysis and simulation results demonstrate that the success rate for the *prospective* reorganization attacks (without the proposer boost mechanism) can reach approximately 99.67% (see Fig. 6).