Blockchain Transaction Censorship: (In)secure and (In)efficient?

Abstract. The ecosystem around blockchain and Decentralized Finance (DeFi) is seeing more and more interest from centralized regulators. For instance, recently, the US government placed sanctions on the largest DeFi mixer, Tornado.Cash (TC). To our knowledge, this is the first time that centralized regulators sanction a decentralized and open-source blockchain application. It has led various blockchain participants, e.g., miners/validators and DeFi platforms, to censor TC-related transactions. The blockchain community has extensively discussed that censoring transactions could affect users' privacy.

In this work, we analyze the efficiency and possible security implications of censorship on the different steps during the life cycle of a blockchain transaction, i.e., generation, propagation, and validation. We reveal that fine-grained censorship will reduce the security of block validators and centralized transaction propagation services, and can potentially cause Denial of Service (DoS) attacks. We also find that DeFi platforms adopt centralized third-party services to censor user addresses at the frontend level, which blockchain users could easily bypass. Moreover, we present a tainting attack whereby an adversary can prevent users from interacting normally with DeFi platforms by sending TC-related transactions.

1 Introduction

On August 8th, 2022, the US Treasury's Office of Foreign Assets Control (OFAC) placed sanctions [17, 18] on the largest zero-knowledge proof (ZKP) mixer, Tornado.Cash (TC) [1], due to alleged facilitation of money laundering. TC has been used to process more than 7B USD worth of cryptocurrencies since its creation in 2019. OFAC added the TC website and related blockchain addresses to the "Specially Designated Nationals And Blocked Persons" (SDN) list. According to the sanctions, US citizens are no longer legally allowed to use the TC website or involve any property or interest transactions with those blacklisted addresses. To our knowledge, this is the first time that centralized regulators sanction a decentralized and open-source blockchain application.

The sanctions have led to a series of consequences. For instance, the largest prior-merge Ethereum mining pool, Ethermine, stopped processing any TC deposit and withdrawal transactions since August 9th, 2022 [3]. Many DeFi platforms, e.g., Uniswap [19], Aave [2], and dYdX [5], have started banning addresses that receive transactions from TC after the sanctions were announced. Centralized transaction propagation services, e.g., Front-running as a Service (FaaS) such as Flashbots, also ban transactions calling OFAC-blacklisted addresses.

Circle, the issuer of the stablecoin `USDC`, has already frozen all `USDC` held in OFAC-blacklisted TC addresses [3].

However, the sanctions of an open-source DeFi application operating on top of blockchains, bring up new questions and extensive discussions in the blockchain community [6] and even within the US government [16]. Privacy advocates argue that the banning of ZKP mixers violates citizens' right to privacy, and OFAC exceeds its statutory authority by treating an autonomous and decentralized application as an individual or entity.

In this paper, we study blockchain censorship from a novel perspective. We investigate if it is possible to achieve "fine-grained" censorship on permissionless blockchains to fully ban tainted transactions. We analyze the censorship during the life cycle of blockchain transactions (i.e., generation, propagation, and validation) to reveal the efficiency and security implications of censoring transactions and addresses. We summarize our contributions as follows:

1. **Censorship Reduces Miners' Security:** We investigate how blockchain miners censor transactions. Our results indicate that users can easily bypass miners' current censorship. Therefore, we propose a fine-grained censoring algorithm. However, we prove that censorship will reduce miners' security, because an adversary can craft tainted transactions to attack miners. We show that the attack comes at zero gas fees when all miners adopt censoring.
2. **Dissect FaaS Censorship Mechanism:** We analyze the blockchain transaction censorship during the propagation process in FaaS. By analysing the relayed blocks by six FaaS (i.e., Flashbots, Eden Network, Manifold, Aestus, Agnostic Gnosis and Blocknative), we find that only Flashbots is complying with OFAC regulations by censoring TC-related transactions. However, our analysis indicates that fine-grained censorship will also reduce FaaS's security.
3. **Bypassing DeFi Platform Censorship:** We analyze how DeFi platforms (e.g., Uniswap, dYdX, and Aave) ban user addresses. We find that DeFi platforms leverage centralized third-party services to censor user addresses at the frontend level. Therefore, users can resort to using intermediary addresses or a command line interface (CLI) to bypass the censorship. Additionally, we present an attack whereby an adversary can deliberately taint innocent addresses by sending transactions involving blacklisted addresses. This attack can prevent users from interacting normally with DeFi platforms' frontend.

2 Background

2.1 Blockchain and Smart Contracts

Blockchains [12,23] are distributed ledgers on top of a global peer-to-peer (P2P) network. Users can join and leave the network freely. There is no central authority to guarantee common agreement on the distributed ledgers among users. Users can achieve agreement through consensus protocols, such as Proof-of-Work (PoW) for prior-merge Ethereum, and Proof-of-Stake (PoS) for post-merge Ethereum. Smart contracts are quasi-Turing-complete programs which can be

executed within a virtual machine. Users can leverage smart contracts to build DeFi services [4,21]. Transactions are propagated over a public P2P or a private relay network. Miners and non-mining traders can manipulate the transaction order and front-run other traders by unilaterally determining the order or paying higher transaction fees to extract Blockchain Extractable Value (BEV) [14].

2.2 Centralized Transaction Propagation Services

Independent of the P2P network, emerging centralized relay services, i.e., FaaS, offer an alternative option for users to bid for the priority to extract BEV by communicating to miners/validators privately. For example, on Flashbots, traders can add an arbitrary number of signed transactions (including transactions from other parties) to their bundle, along with metadata specifying the bundle execution logic. Traders can then submit transaction bundles directly to miners/validators without a broadcast on the P2P network. Auctions through Flashbots are risk-free, meaning unsuccessful bids do not need to pay transaction fees.

2.3 ZKP Mixers

ZKP mixers, inspired by Zerocash [15], are one of the most widely-used privacy solutions for non-privacy-preserving blockchains. ZKP mixers are running on top of smart-contract-enabled blockchains, e.g., Ethereum. Upon using a mixer, a user deposits a fixed denomination of coins into a pool and later withdraws these coins to another address [1, 9, 20]. When used properly, ZKP mixers can break the linkability between addresses, thus enhancing users' privacy. Therefore, ZKP mixers, e.g., TC [1], are widely used for money laundering [20] and receiving the initial funds to launch on-chain attacks [24].

2.4 Blockchain Regulation and Censorship

Although permissionless blockchains, such as Bitcoin and Ethereum, seem to be able to evade regulation and censorship through their decentralization, their surrounding ecosystem has attracted interest from regulators [8, 11]. Regulators have started enforcing existing financial regulations for off-chain services, such as anti-money laundering (AML) regulations for centralized exchanges. Though off-chain regulations will not directly affect on-chain activities, blockchain participants may follow regulations to ban transactions related to specific addresses. For instance, on August 8th, 2022, the US OFAC announced sanctions against TC, and added TC-related addresses to the SDN List [17, 18]. To the best of our knowledge, this is the first time that centralized regulators sanction a decentralized application. After the announcement, some Ethereum miners, FaaS, and DeFi platforms have started censoring TC-related transactions and addresses [3].

3 System Model

In this section, we outline our system and threat model for blockchain censoring.

3.1 System Components

Address: On permissionless blockchains, a user has at least one public/private key-pair, which corresponds to their *address* and controls cryptocurrencies.
Smart Contract: Smart contracts are quasi-Turing-complete programs that typically execute within a virtual machine. A smart contract function can be called by a transaction, and can also call the functions of other contracts. A smart contract can emit *events* when successfully being executed.
Transaction: To transfer assets or trigger the execution of smart contract functions, a user signs a transaction with its private key. The transaction's sender pays for the cost of the triggered smart contract execution, i.e., transaction/gas fees. An *internal transaction* is a transaction triggered by a smart contract as a result of one or more previous transactions.
Block: A block includes a list of transactions. A blockchain consists of a growing list of blocks, which are securely linked together using cryptographic hashes.
Miners/Validators: Miners on PoW blockchains, or validators on PoS blockchains, are responsible for: *(i)* sequencing transactions, i.e., specifying the order of transactions within a block; *(ii)* verifying transactions and blocks; *(iii)* confirming transactions and proposing blocks; and *(iv)* propagating data. In this paper, we regard the terms "miner" and "validator" as interchangeable.
Blockchain Network: Blockchains are operating on top of a global P2P network. Users can join, exit, and discover other nodes in the network. A transaction can be propagated over the network. Moreover, users can leverage centralized transaction propagation services, i.e., FaaS, to transmit a transaction directly to miners, without broadcasting it to the remaining network.

3.2 Blockchain Censoring

We introduce the fundamental components of blockchain censoring as follows.
Blacklisted Addresses: Blacklisted addresses are a list of blockchain addresses (including smart contract addresses) that are banned by off-chain regulators (e.g., the US OFAC). Users are legally prohibited from involving any property or interest transactions with those addresses.
Tainted Transactions: We define a transaction as tainted if it *(i)* is issued by a blacklisted address, *(ii)* transfers assets to a blacklisted address, or *(iii)* triggers a function of a blacklisted smart contract address.
Transaction Censorship: We define *transaction censorship* as an action to prevent a tainted transaction from being generated, propagated, or validated.
Censorship Categories. Fig. 1 shows the life cycle of a transaction. A transaction can be censored at different steps (i.e., generation, propagation, and validation). In the following, we list the various censorship categories.
- *Generation censoring*: A censoring blockchain application (e.g., Centralized Exchange or DeFi platform) will ban the interaction between their frontend and the addresses which *(i)* attempt to generate tainted transactions, or *(ii)* interacted with blacklisted addresses.

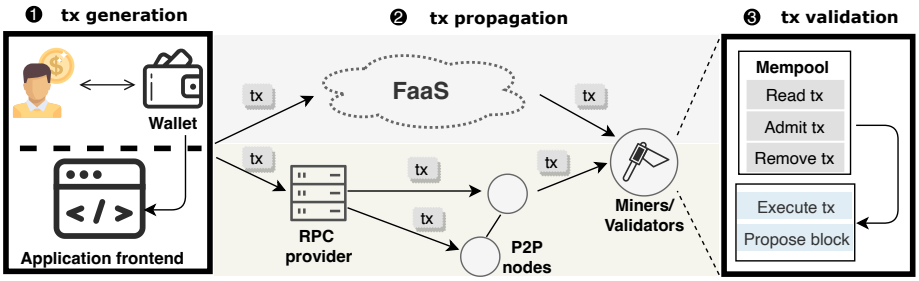

Fig. 1: Blockchain transaction life cycle. In step ❶, a user's wallet creates a transaction tx, which involves generating a signature with the user's private key. This generation can be performed locally or interact with a blockchain application's frontend. In step ❷, the wallet sends tx to a RPC provider, which will broadcast tx into the entire P2P network. The wallets can also send tx to a FaaS, which will forward tx to validators via a private network. In step ❸, upon receiving tx, a validator will first collect the transaction into its mempool, and will then include tx into a newly proposed block after verifying tx. A transaction can be censored in the steps of generation, propagation, or validation.

- *Propagation censoring*: A censoring FaaS or a P2P node will not choose to forward any tainted transactions to miners/validators or other P2P nodes.
- *Validation censoring*: A censoring miner/validator will not include tainted transitions in their proposed blocks. However, a censoring miner/validator can receive blocks which contain tainted transactions and are proposed by others.

3.3 Threat Model

Given a censoring blockchain participant, which can be a miner/validator, FaaS, or application, the adversary's goal is to *(i)* bypass the participant's censorship on tainted transactions, or *(ii)* attack the participant to prevent it from executing or forwarding non-tainted transactions.

We further assume that the adversary possesses the following capabilities:
Sending Private Transactions: The adversary can send an unconfirmed transaction directly to miners or FaaS via their private RPC, or propagate the transaction over the remaining P2P network.
Crafting Complicated Transactions: The adversary can create a tainted transaction in which multiple contracts are called. The time for executing the transaction increases over the number of contracts.

4 Censorship During Transaction Validation

In the following, we investigate miners' censorship, and propose a DoS attack against the censoring miners through crafting sophisticated transactions.

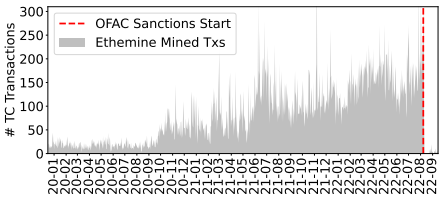

Fig. 2: TC deposit and withdrawal transactions mined by Ethermine over time. Ethermine stopped processing TC transactions during August 10th and August 24th, 2022.

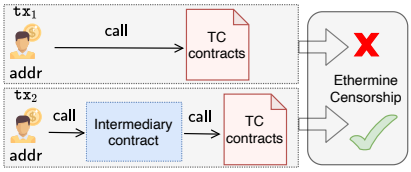

Fig. 3: Ethermine bans the transaction that calls TC contracts directly (e.g., tx_1). However, if a user leverages an intermediary contract to interact with TC, then the transaction (e.g., tx_2) will pass Ethermine's censorship.

4.1 Miners' Censorship on Tainted Transactions

The OFAC sanctions against TC have had an influence on Ethereum miners. As shown in Fig. 2, we plot the distribution of TC transactions mined by the largest mining pool, Ethermine, before the Ethereum merge, i.e., September 15th, 2022. We observe that, from August 10th to August 24th, 2022, Ethermine stopped processing any transactions related to deposits and withdrawals in TC (cf. Fig. 2). This indicates that Ethermine censors TC-related transactions. Interestingly, we find that 1 deposit and 98 withdrawal transactions can still bypass Ethermine's censorship after August 24th, 2022.

Miners' Censorship on TC. To understand how Ethermine censors TC-related transactions, we perform the following experiments and analysis.
- We create a transaction tx_1 that calls TC contracts directly, and send tx_1 to Ethermine through its private RPC[1]. We then observe that tx_1 will not be mined.
- We analyze the 99 transactions which bypassed Ethermine's censorship after August 24th, 2022. We find that all these transactions first call an intermediary contract, which is not a blacklisted address, and the intermediary contract will later call blacklisted TC contracts through internal transactions.

As shown in Fig. 3, we can thus infer that, Ethermine's censorship on blacklisted addresses works as follows: *(i)* Upon receiving a pending transaction tx, Ethermine checks tx's from-address and to-address. *(ii)* If the from-address or to-address of tx is an OFAC-blacklisted address, i.e., a blacklisted address is directly called in tx, then Ethermine evicts tx from its mempool for pending transactions.

Improving Miners' Censorship Mechanism. Fig. 3 depicts that Ethermine does not censor the transactions that call TC contracts through internal transactions, which means users can still interact with TC using intermediary contracts. Generally, blacklisted addresses can be called through internal transactions, and a censoring miner has to execute a transaction to check if it is tainted. We thus propose the following claim (cf. Claim 1).

[1] https://ethermine.org/private-rpc, available on September 1st, 2022

Algorithm 1: Fine-grained censorship on tainted transactions.

Input: tx: an unconfirmed transaction
Output: True or False: tx will be mined or not
Param : $\{\texttt{addr}\}_{ban}$: set of blacklisted addresses

1 **if** tx*'s* from-address $\in \{\texttt{addr}\}_{ban}$ *or* tx*'s* to-address $\in \{\texttt{addr}\}_{ban}$ **then**
2 | **return** False
3 Execute tx locally, and record the set of called addresses, $\{\texttt{addr}\}_{call}$.
4 **if** $\{\texttt{addr}\}_{call} \cap \{\texttt{addr}\}_{ban} \neq \emptyset$, **then return** True; **else return** False

Claim 1 *Given an unconfirmed transaction* tx*, to check if it is tainted, a censoring node can simply simulate* tx *locally.*

Based on Claim 1, we propose a novel algorithm to check tainted transactions. As shown in line 1-3 of Alg. 1, we keep Ethermine's style of censorship as the first step to filter the transactions from a blacklisted address or calling blacklisted addresses directly. Moreover, given a transaction tx, to check if any blacklisted addresses are called through internal transactions in tx, miners execute tx locally to extract all called addresses and check whether they are blacklisted (cf. line 1-4 in Alg. 1). Therefore, Alg. 1 can censor all tainted transactions.

4.2 DoS Censoring Miners through Crafting Tainted Transactions

In the following, we investigate the downside of the improved censorship algorithm (cf. Alg. 1), which can enable an adversary to attack censoring miners.

Censoring Computation Cost. We craft a TC-related transaction with multiple intermediary contracts (cf. Fig. 4), and the TC contract is called by the last contract. For each intermediary contract, we add time-consuming operations (e.g., Line 11 in Fig. 16). We then evaluate the crafted transactions' execution time when calling a various number of intermediary contracts. We adopt Hardhat to deploy the contracts and execute the transactions locally on an Ethereum Erigon node. The node is running on a macOS Ventura machine with an Apple M1 chip with 8-core CPU, 8-core GPU, 16-core Neural Engine, 16 GB of RAM, and 2 TB SSD storage in configuration. Fig. 5 shows that the execution time approximately increases linearly over the number of intermediary contracts.

DoS Censoring Miners. The expensive censoring computation cost brings up the opportunities for the adversary to attack miners. Intuitively, the adversary can craft numerous complicated and tainted transactions, and keep sending them to the victim miner. Therefore, the victim will be exhausted from censoring those complex transactions and cannot process new non-tainted transactions.

Attack Strategy. We propose the following strategy to DoS censoring miners: *(i)* The adversary crafts m complicated and tainted transactions. Each of those transactions is configured with a high gas price, which is *much higher* than any existing transactions in the victim's mempool. *(2)* The adversary then keeps sending the m crafted transactions to the victim, via the victim's private RPC. We provide the attack results in a private Ethereum network in Appendix A.

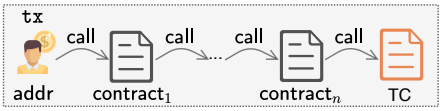

Fig. 4: In a crafted TC transaction, n intermediary contracts are called consecutively, and the final intermediary contract will call TC contracts.

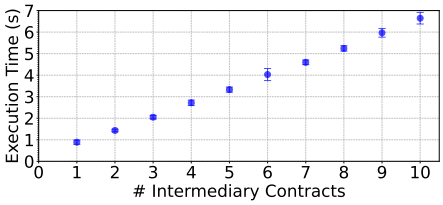

Fig. 5: A crafted transaction's execution time on a local node when calling multiple intermediary contracts.

4.3 Attack Cost

We analyze the attack costs when DoSing censoring miners in different cases.

Case 1: All Miners Adopt Censoring. If all miners adopt censoring, then none of the tainted transactions will be mined. Therefore, the adversary does not need to pay any transaction fees. The attack cost will only be the cost of buying attack machines, electricity, network bandwidth consumption, etc. We assume that these costs are constant and denote their sum as $\mathcal{C}_{cnst}$.

Attacking All Miners Simultaneously. As shown in Fig. 6, instead of attacking a specific miner, the adversary can DoS all censoring miners simultaneously. The adversary can broadcast a crafted and tainted transaction tx to the entire P2P network (rather than send tx via a specific miner's private RPC), and every miner will finally receive tx. In this case, all censoring miners will waste their computation power on checking tx but will not mine it. If the adversary broadcasts sufficient tainted transactions with a high gas price, then all censoring miners could be DoSed. Note that the attack cost comes at zero transaction fees because no tainted transactions will be mined.

Case 2: Non-Zero But Not All Miners Adopt Censoring. In this case, some miners do not choose to adopt censoring, and a tainted transaction might be successfully mined. The adversary might suffer from the cost of gas fees.

As shown in Fig. 7, when a censoring miner receives a transaction tx and finds that tx is tainted, then the miner can forward tx to its peers. Finally, a non-censoring miner will receive tx and mine it. Therefore, the adversary has to pay the transaction fee of tx.

Attacking A Single Censoring Miner. Assume that the average fee for a crafted transaction tx is f. If the censoring miner forwards tx to its non-censoring peers, then the adversary needs to pay $f \cdot m + \mathcal{C}_{cnst}$. Recall that m is the number of transactions that the adversary sends to the miner. However, if the censoring miner just abandons the crafted transactions, then the attack cost is $\mathcal{C}_{cnst}$.

Attacking Multiple Censoring Miners Simultaneously. The adversary can also attempt to broadcast the tainted and complicated transactions to the entire P2P network. Analogously, the cost of transaction fees is determined by the number m of crafted transactions and the average transaction fee f, i.e., $f \cdot m$. Therefore, the

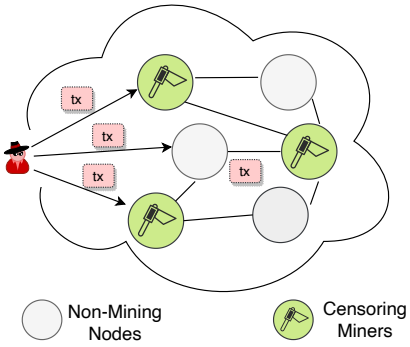

Fig. 6: Attacking all miners simultaneously when all miners adopt censorship. No tainted transaction will be mined, and the adversary does not need to pay any transaction fees.

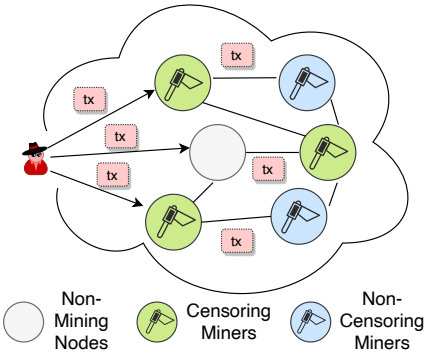

Fig. 7: Attacking multiple censoring miners simultaneously when there are non-censoring miners. The adversary will pay a cost if tainted transactions are mined by non-censoring miners.

total attack cost is $cost = \mathcal{C}_{cnst} + f \cdot m$. Moreover, a crafted transaction tx might be first mined by a non-censoring miner before any censoring miners receive it. Although censoring miners will not pre-execute tx to check if it is tainted, they still need to spend computation source on validating the block including tx.

5 Censorship During Transaction Propagation

This section analyzes FaaS's censorship during transaction propagation.

5.1 FaaS Workflow

We take Flashbots [22] as an example to analyze the workflow of FaaS. Fig. 8 describes the transaction order flows before the Ethereum merge. ❷ denotes the public user order flow where a user sends its transactions to a public node (e.g., Infura) RPC. In contrast, ❶ shows the private user order flow, where a user switches to a private RPC endpoint to protect its transaction from being front-/back-run by adversaries [4]. ❸ depicts the searcher order flow, where a searcher listens to public transactions propagated over the P2P network. Once finding BEV opportunities, the searcher constructs a bundle of transactions in an immutable order and submits it to the Flashbots relay.

However, several important changes happened after the Ethereum merge. Specifically, Flashbots implements a protocol named Proposer–Builder Separation (PBS) via *MEV-Boost*, which separates the block construction role from the block proposal role. PBS allows validators (i.e, *proposers*) to outsource the block-building roles to specialized parties called *builders*. As shown in Fig. 9, searchers

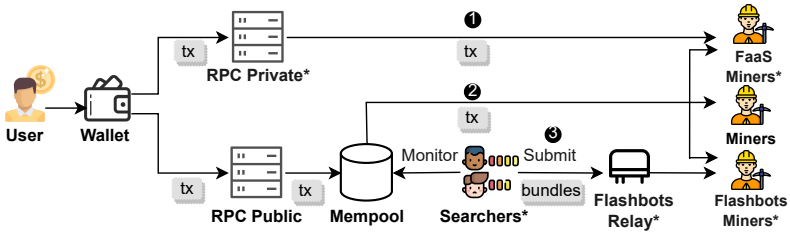

Fig. 8: Flashbots transaction order flows pre-PBS. ❶,❷ and ❸ denote private, public and search order flows. * denotes entity with potential censoring power.

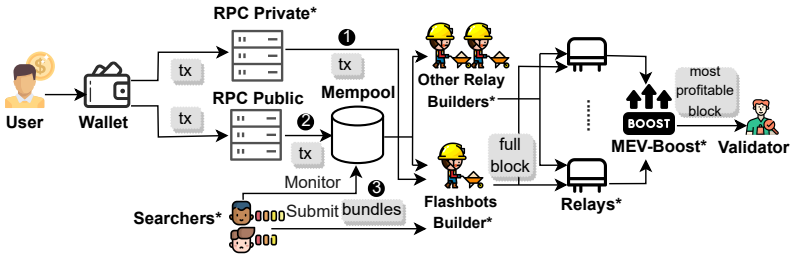

Fig. 9: Flashbots transaction order flows post-PBS. ❶,❷ and ❸ denote private, public and search order flows. * denotes entity with potential censoring power.

submit bundles to builders, which are responsible for building full blocks with available transactions and submitting bids to relays. A relay verifies the validity of the execution payload and selects the most-profitable block sent by all connected builders, while the MEV-Boost picks the best block from multiple relays. The block proposer receives blind blocks, signs the most profitable block, and sends it back to the relay. Once verifying the proposer's signature, the relay responds with the full block for the proposer to propose to the network. Note that although MEV-Boost mitigates the centralization of the validator set, it may cause the builder centralization, e.g., the dominant builder with the highest inclusion rate may receive exclusive order flows from users and searchers [7].

5.2 FaaS Censorship Mechanism

To understand how FaaS adopts censorship, we conduct empirical analysis by crawling all blocks relayed by six FaaS through their public APIs, i.e., Flashbots, Eden Network, Manifold, Aestus, Agnostic Gnosis and Blocknative from the Ethereum block 15,537,940 (September 15th, 2022) to 16,331,031 (January 4th, 2023). We also crawl the transactions in which TC deposit or withdrawal events are emitted. As shown in Figures 10 and 11, we plot the total number of relayed blocks, and the relayed blocks which include TC-related transactions. We observe that, although Flashbots relay the most blocks during the timeframe (cf. Fig. 10),

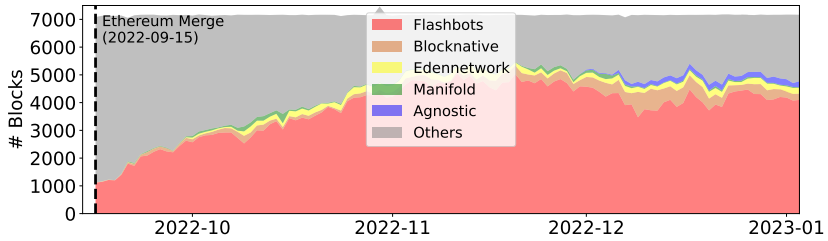

Fig. 10: Distribution blocks relayed by different FaaS after the Ethereum Merge. Flashbots relay the most blocks, i.e., more than 52.54% of the total blocks during September 15th, 2022 and January 4th, 2023 are relayed by Flashbots.

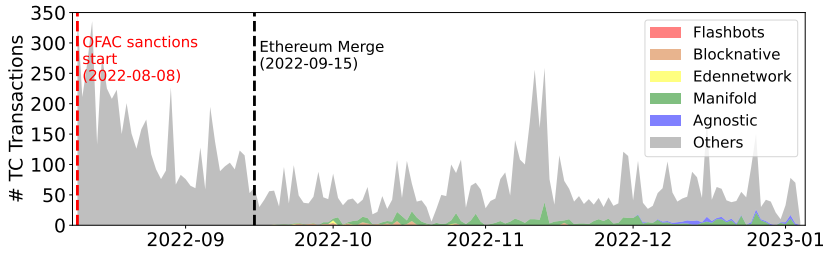

Fig. 11: Distribution of TC deposit and withdrawal transactions from August 8th, 2022 to January 4th, 2023. Flashbots do not relay any blocks including TC-related transactions after the Ethereum block 15537940 (September 15th, 2022).

none of TC-related transactions are relayed by Flashbots (cf. Fig. 11). This result indicates that Flashbots ban TC deposit and withdrawal transactions.

We take Flashbots as an example to dissect its censorship. After checking the code in its GitHub repositories, we find that Flashbots complied with OFAC regulations by censoring TC-related transactions in the following ways:

Period 1: From OFAC sanction announcement to the merge. First, the Flashbots RPC endpoint censors the user private order flow (i.e., ❶ in Fig. 8). It configures the blacklisted TC-related addresses, and checks whether the to-address or from-address in the transaction are blacklisted. The transaction will be censored if any blacklisted address is found. In addition, the Flashbots Relay censors the searcher order flow (i.e., ❸ in Fig. 8) by checking the blacklisted addresses in the received bundles. It is worth noting that a searcher or miner in Fig. 8 may also censor transactions by simulating the transaction execution.

Period 2: Post merge. Additionally, Flashbots has a Block Validation Geth client to help censor TC-related transactions. In contrast to the Flashbots RPC endpoint and the Flashbots Relay, which simply check whether the TC contracts are called directly, the Block Validation Geth client checks all the intermediary contract calls in a given transaction execution trace, i.e., adopting the fine-grained censorship in Alg 1. Similarly, a builder or searcher in Fig. 9 still has

potential censoring power. In contrast, a validator can not censor transactions since it receives a blind block from the MEV-Boost.

5.3 DoS Censoring FaaS Searchers and Builders

In the following, we discuss the potential DoS attacks against censoring searchers and private RPCs in the post-merge FaaS system based on the following assumptions: *(i)* the adversary is the user in Fig. 9; *(ii)* searchers and builder are honest because they have to maintain their reputations; and *(iii)* the censoring entities perform fine-grained check (cf. Alg. 1) on blacklisted addresses.

Attack Strategy. Similar to attacking a censoring miner in Section 4.2, the adversary can create numerous tainted transactions by adding a call to a blacklisted address after calling several intermediary contracts consecutively (cf. Fig. 4). Based on the assumption *(iii)*, an adversary controlling multiple addresses can launch a targeted attack against a given private RPC, or even a non-targeted attack against searchers. Note that we do not discuss the possibility of attacking relays, as we assume that searchers and builders are honest.

6 Censorship During Transaction Generation

In this section, we analyze DeFi platform censorship during the transaction generation process. For the completeness, we refer the reader to Appendix B for other blockchain components' censorship.

6.1 Non-Transparent Frontend-Level Censorship

Although DeFi protocols are running on a decentralized blockchain, the websites interacting with the protocols are centralized. The entities that develop and maintain the websites will risk breaking the law if they do not follow the OFAC sanctions. However, the centralized entities' censorship is non-transparent (cf. Fig. 12). In the following, we leverage public information to analyze the platforms' censorship, which claims to follow the OFAC sanctions.

Uniswap. Uniswap is a Decentralized Exchange running on top of Ethereum. Uniswap claims that they cooperate with TRM Labs to identify on-chain financial crime and block addresses that are owned or associated with clearly illegal behaviors such as sanctions, terrorism financing, hacked or stolen funds, etc [19]. Therefore, the OFAC-sanctioned TC addresses are also banned by Uniswap Decentralized Application. Although it was reported that Uniswap has prohibited 253 addresses on its frontend, so far as we understand, Uniswap and TRM Labs do not intend to publish their censoring mechanism and data.

Aave. Aave is an on-chain lending platform. Similar to Uniswap, Aave leverages TRM Labs to determine financial crime and other prohibited activities [2]. Although Aave provides an API for users to check whether their addresses will be banned, Aave does not disclose the censorship details on their IPFS frontend.

Table 1: TC user addresses interacting with DeFi platforms from September 15th, 2022 to January 4th, 2023.

Platform	Censorship Start Date	TC User Addresses	
		depositors	withdrawers
Uniswap	Before 2022/08/23 [19]	88	213
Aave	2022/08/10 [2]	2	5
dYdX	2022/08/10 [5]	1	0

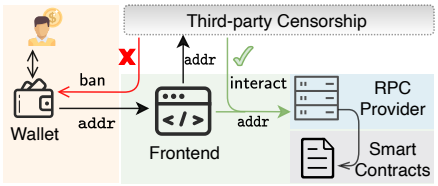

Fig. 12: DeFi platforms generally leverage a third-party service to determine whether an address should be banned.

dYdX. dYdX is a DeFi platform supporting perpetual, margin and spot trading, as well as lending and borrowing. dYdX has confirmed that it blocked several addresses in line with the OFAC's sanctions against TC [5]. dYdX claims they have long utilized "compliance vendors" to identify sanction-related addresses. However, dYdX's censorship is still not public at the time of writing.

6.2 Investigating DeFi Platforms' Censorship

To investigate how a DeFi platform censors blacklisted addresses, we leverage on-chain data to analyze if TC user addresses can still interact with the DeFi platform after the OFAC sanctions are announced.

Post-Sanction TC User Addresses. We crawl the addresses that are used to deposit and withdraw in the four TC `ETH` pools after the OFAC sanctions are announced. We identify 2,282 TC user addresses during September 15th, 2022 and January 4th, 2023, out of which 805 are used to deposit and 1,581 to withdraw. For these 2,282 addresses, we crawl their historical transfers of `ETH` and ERC20 tokens. We also crawl 379 labeled addresses from Etherscan of the three censoring DeFi platforms. We finally analyze whether the 2,282 TC user addresses interact with the platforms after they deposit/withdraw into/from TC.

Results. As shown in Table 1, we identify that 89 deposit and 216 withdrawal addresses can still interact with the three censoring platforms. For instance, on October 29th, 2022, the address 0x2d7...7F7 withdrew from TC 100 `ETH` pool at block 15,853,770 and then swapped 54.5 `ETH` to 88,509 `USDC` on Uniswap at block 15,853,783. These results indicate that the existing DeFi platforms' censorship mechanism is inefficient and may cause false negatives.

6.3 Tainting Innocent Addresses

In the following, we will show that even if a DeFi platform can *perfectly* ban TC users, the censorship will cause new security issues. Specifically, we discuss a tainting attack, where the adversary can leverage TC withdrawals to taint innocent addresses. This attack can cause the victim addresses to be blocked when interacting with the frontend of censoring DeFi platforms (cf. Fig. 13).

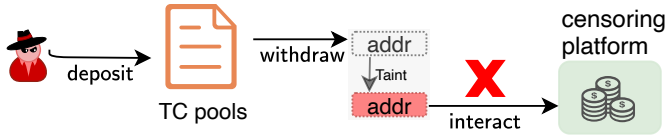

Fig. 13: Tainting attack overview. After depositing into TC, the adversary assigns an innocent address as the withdrawal address. Then the address will be tainted, and thus blocked when interacting with the censoring DeFi platforms' frontend.

Attack Strategy. Given a victim address `addr`, the adversary performs the following steps to taint `addr` and block `addr`'s activities on censoring platforms.
- 1. The adversary deposits coins into a mixer pool (e.g., TC 0.1 `ETH` pool).
- 2. Upon withdrawing coins from the pool, the adversary assigns the victim address `addr` as the withdrawal address.
- 3. The victim address `addr` will receive the assets from the pool; therefore, `addr` will be banned when interacting with censoring DeFi platforms.

Attack Cost. The attack cost is affected by the selected mixer pool. The cost of tainting an address equals the minimum denomination that the pool supports. The overall cost also increases linearly with the number of tainted addresses.
Attack Consequences. The tainting attack will lead the victim addresses to be banned when interacting with the censoring DeFi platforms' frontend. The ban could also cause utilization problems for DeFi users. For instance, on Aave, blocked user addresses with active loans will not be able to access their borrowing position via the frontend and manage the position health to avoid being liquidated [21]. We provide an example to show how an adversary can benefit from tainting a user address on a censoring lending platform as follows.
- 1. Consider a user address `addr`, which supplies `ETH` and borrows `DAI` in Aave lending pool. The adversary performs the tainting attack against `addr`.
- 2. The adversary then leverages flash loans [24] to manipulate the price of `DAI`, which will cause the victim's borrowing position to become unhealthy.
- 3. As the victim is blocked by the Aave frontend and cannot access the position in time, the adversary can liquidate the unhealthy position to gain profits.

6.4 Bypassing Frontend-Level Censorship

In the following, we propose two methods to bypass frontend-level censorship.
Interacting with Smart Contracts via CLI. DeFi users can interact with the platform smart contracts through a CLI or by forking the platform project to create their own frontend interface. As shown in Fig. 12, in this way, there will be no third-party censorship, and user addresses will not be banned. However, this method might be beyond the technical knowledge of many DeFi users.
Leveraging Intermediary Addresses. Another method is to adopt a non-tainted address to interact with censoring DeFi platforms. To do so, users need to transfer their assets from their tainted addresses to non-tainted ones. For instance, we observe that a TC user transfers the withdrawn `ETH` to a non-tainted

address via an intermediary address, to swap ETH to renBTC on Uniswap, i.e., $TC \xrightarrow{49.8\ \texttt{ETH}} \texttt{addr}_0 \xrightarrow{25.3\ \texttt{ETH}} \texttt{addr}_1 \xrightarrow{16.5\ \texttt{ETH}} \texttt{addr}_2 \xrightarrow{11.97\ \texttt{ETH}} Uniswap \xrightarrow{0.94\ \texttt{renBTC}} \texttt{addr}_2$. In this way, the non-tainted address $\texttt{addr}_2$ is not blocked by Uniswap.

7 Related Work

Blockchain Censorship. Moser *et al.* [11] discuss how transaction blacklisting would change the Bitcoin ecosystem and how it can remain effective in the presence of privacy-preserving blockchains. Kolachala *et al.* [8] investigate the blacklisting technique to combat money laundering, and point out that there are unanswered questions and challenges with regard to its enforcement.

Money Laundering on Blockchains. Wang *et al.* [20] investigate how users leverage ZKP mixers, e.g., TC and Typhoon.Network to launder money. Wang *et al.* also propose heuristics to link mixer deposit and withdrawal addresses, which can be used to trace mixer users' coin flow. Zhou *et al.* [24] indicate that DeFi attackers can receive their source of funds from mixers to launch attacks. Their results show that 55 (21%) and 12 (4.6%) of the 181 attack funds originate from the ZKP mixers on ETH and Binance Smart Chain, respectively.

Blockchain DoS Attacks. Li *et al.* [10] propose a series of low-cost DoS attacks named DETER, which leverages Ethereum clients' vulnerability in managing unconfirmed transactions. DETER can disable a remote Ethereum node's mempool and deny the critical downstream services in mining and transaction propagation. Perez *et al.* [13] present a DoS attack, called Resource Exhaustion Attack, which systematically exploits the imperfections of the Ethereum metering mechanism to generate low-throughput contracts.

8 Conclusion

This paper studies the security implications of blockchain transaction censorship. Specifically, we show that miners or validators can execute the transaction to censor whether blacklisted addresses are called. This additional execution requirement enables an attack whereby an adversary could deliberately DoS a censoring miner or validator through crafting numerous tainted and complicated transactions. Our analysis shows that the attack comes at zero transaction fees when all miners or validators adopt censoring. Moreover, we find that a censoring FaaS might also suffer from such an attack. Furthermore, we show that current DeFi platforms' censorship is at the frontend level, and users can efficiently bypass the censorship using CLI or intermediary addresses. We hope our work can engender further research into more secure solutions for blockchain censorship.

References

1. Tornado cash. Available at: https://tornado.cash/, before August 8th, 2022.
2. Aave. Address screening. https://docs.aave.com/faq/#address-screening.

3. Chainalysis. Understanding tornado cash, its sanctions implications, and key compliance questions. https://blog.chainalysis.com/reports/tornado-cash-sanctions-challenges/.
4. Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020.
5. dydx. Tornado outage. https://dydx.exchange/blog/tornado-outage.
6. Brito Jerry and Van Valkenburgh Peter. Analysis: What is and what is not a sanctionable entity in the tornado cash case. 2022.
7. Quintus Kilbourn. Order flow, auctions and centralisation. The Science of Blockchain Conference, 2022.
8. Kartick Kolachala, Ecem Simsek, Mohammed Ababneh, and Roopa Vishwanathan. Sok: Money laundering in cryptocurrencies. In *The 16th International Conference on Availability, Reliability and Security*, pages 1–10, 2021.
9. Duc V Le and Arthur Gervais. Amr: Autonomous coin mixer with privacy preserving reward distribution. *Advances in Financial Technologies (AFT'21)*, 2021.
10. Kai Li, Yibo Wang, and Yuzhe Tang. Deter: Denial of ethereum txpool services. In *Proceedings of the 2021 ACM CCS*, pages 1645–1667, 2021.
11. Malte Möser and Arvind Narayanan. Effective cryptocurrency regulation through blacklisting. *Preprint*, 2019.
12. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
13. Daniel Perez and Benjamin Livshits. Broken metre: Attacking resource metering in evm. In *Proceedings of the 27th NDSS*. Internet Society, 2020.
14. Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *IEEE Symposium on Security and Privacy*, 2022.
15. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.
16. Emmer Tom. Letter to treasury secretary yellen regarding the unprecedented sanctioning of tornado cash, 2022. Available at: https://twitter.com/RepTomEmmer/status/1562084891247902721.
17. U.S. DEPARTMENT OF THE TREASURY. Cyber-related sanctions, 2022. Available at: https://home.treasury.gov/taxonomy/term/1546.
18. U.S. DEPARTMENT OF THE TREASURY. U.s. treasury sanctions notorious virtual currency mixer tornado cash, 2022. Available at: https://home.treasury.gov/news/press-releases/jy0916.
19. Uniswap. Address screening guide, 2022. Available at: https://support.uniswap.org/hc/en-us/articles/8671777747597-Address-Screening-Guide.
20. Zhipeng Wang, Stefanos Chaliasos, Kaihua Qin, Liyi Zhou, Lifeng Gao, Pascal Berrang, Ben Livshits, and Arthur Gervais. On how zero-knowledge proof blockchain mixers improve, and worsen user privacy. *Proceedings of WWW*, 2023.
21. Zhipeng Wang, Kaihua Qin, Duc Vu Minh, and Arthur Gervais. Speculative multipliers on defi: Quantifying on-chain leverage risks. *FC' 22*, 2022.
22. Ben Weintraub, Christof Ferreira Torres, Cristina Nita-Rotaru, and Radu State. A flash (bot) in the pan: measuring maximal extractable value in private pools. In *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022.
23. Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.
24. Liyi Zhou, Xihan Xiong, Jens Ernstberger, Stefanos Chaliasos, Zhipeng Wang, Ye Wang, Kaihua Qin, Roger Wattenhofer, Dawn Song, and Arthur Gervais. Sok: Decentralized finance (defi) incidents. *arXiv preprint arXiv:2208.13035*, 2022.

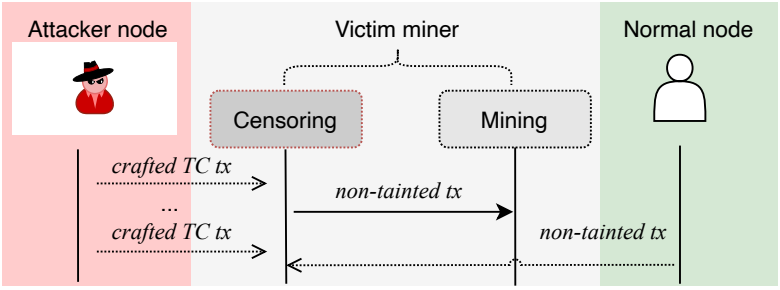

Fig. 14: Attacking a censoring miner through crafted TC transactions.

A Evaluation DoS Attack in a Private Network

Attack Setup. To evaluate the attack's efficiency, we set up a private Ethereum network to perform the attack. As shown in Fig. 14, the private network consists of three nodes: a normal node, an attacker node, and a victim miner. To build the victim miner, we modify the Ethereum Geth execution client to adopt our improved censoring mechanism (cf. Alg. 1) to filter blacklisted addresses-related transactions. We run the victim node on a macOS Monterey machine with an Apple M1 chip (8-core, 3.2 GHz), 8 GB of RAM, and 512 GB SSD storage in configuration. For the normal node and the victim node, we run them respectively on a Kali Linux ARM machine with Raspberry Pi 4 Model B (Quad core, 1.5GHz), 4 GB of RAM, and 256 GB SD card in configuration.

Evaluation Metrics. In our experiment, we configure the normal node to generate non-tainted transactions and send them to the victim miner, at a rate of 4 transactions per second. The attacker node keeps sending crafted and complicated TC transactions to the victim. The process lasts for 400 seconds. We configure the victim node as the only miner with a block generation time of 4 seconds. The metric we use for attack effectiveness is the number of non-tainted transactions included by the victim miner in each block. We set the gas price in the tainted transactions as $2\times$ higher than the one in the non-tainted transactions. We count the cumulative number of transactions in the blocks when the attacker node sends crafted transactions at a rate of 0, 4 and 16 transactions per second. We repeat the experiment by three times and report the average number of cumulative transactions.

Attack Results. Fig. 15 depicts the attack results in a local private network. We observe that the cumulative number of included transactions increases linearly over block numbers. Moreover, the faster the crafted transactions are being sent, the fewer normal transactions can be included in the blocks proposed by the victim node. We recall that the attack costs zero gas fees because none of the crafted transactions will finally be included in the blocks.

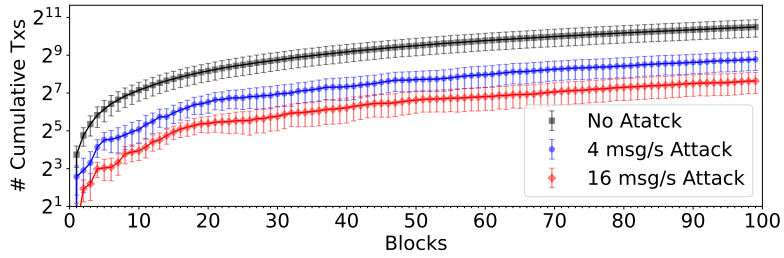

Fig. 15: Number of cumulative transactions in blocks under different attack cases.

```solidity
pragma solidity ^0.8.7;
interface ITornadoRouter {
    function deposit(address _tornado, bytes32 _commitment,
        bytes calldata _encryptedNote) external payable;
}
contract Intermediary{
    ITornadoRouter constant Router = ITornadoRouter(0xd90...);
    address Tc_01_ETH = address(0x12D...);
    function SimpleTransfer(bytes32 commit, bytes calldata
        note) public payable {
        uint256 tmp0 = 123456789;
        uint256 tmp1 = 987654321;
        for (j = 0; j < 10000; j++) { tmp0 += tmp1;}
        Router.deposit{value: msg.value}(Tc_01_ETH, commit,
            note);
    }
}
```

Fig. 16: Intermediary contract code example (in Solidity) for depositing into TC.

B Non-Frontend-Level Censorship

In the following, we briefly discuss non-frontend-level censorship.

Smart Contract Censorship. Circle is a P2P payment technology company, which issues the `USDC` stablecoin. Circle can control the smart contract of `USDC` and configure blacklisted addresses into it. After crawling the `Blacklist` events related to the `USDC` contract, we find that Circle has blacklisted three TC `USDC` pool addresses. This smart contract-level forbiddance has frozen 74,900 `USDC` belonging to TC `USDC` pools.

RPC Provider Censorship. Two famous Ethereum RPC providers, Infura and Alchemy, have blocked their API access for TC [3]. Users cannot connect their wallets to the TC frontend through Infura or Alchemy APIs. However, the two RPC providers' censorship is limited to the frontend interface, users can still interact with TC contracts using CLI.