Temporary Censorship Attacks in the Presence of Rational Miners

Fredrik Winzer
Corporate Research
Robert Bosch GmbH
Stuttgart, Germany
fredrik.winzer@de.bosch.com

Benjamin Herd
Corporate Research
Robert Bosch GmbH
Stuttgart, Germany
benjamin.herd@de.bosch.com

Sebastian Faust
Department of Computer Science
Technical University Darmstadt
Darmstadt, Germany
sebastian.faust@cs.tu-darmstadt.de

***Abstract*—Smart contracts allow for exchange of coins according to program rules. While it is well known that so called bribery contracts can influence the incentive mechanism of a Nakamoto-style consensus, we present a more fine-grained bribery attack incentivizing a temporary censorship against a specific account. To this end, we introduce three different bribery contracts on the blockchain where each uniquely manipulates the rewards that a rational miner would receive. Additionally, we formalize the established bribery mechanisms as a Markov game and show for each game the existence of equilibria leading to successful censorships. Finally, we compare the bribery mechanisms with respect to the scalability of the attack costs and the strategic dominance. Our work is motivated by off-chain protocols including payment and state channels which require to publish transactions within a certain amount of time. In such off-chain protocols a temporary censorship attack can result into significant financial damage.**

***Index Terms*—smart contract, bribery, censorship, mechanism**

I. Introduction

Distributed ledger technologies such as Bitcoin [17] and Ethereum [6] enable decentralized money transfer and execution of so-called smart contracts, thereby enabling a wide range of new applications like decentralized autonomous organizations and payment networks. The security of these applications is based on the correctness of the ledger which contains records of all processed transactions and therefore account states. To extend the ledger, the miners run a consensus protocol, e.g. by competing in solving a cryptographic puzzle. In technologies such as Ethereum, each correct extension of the blockchain rewards the miner with a block reward.

Moreover, the miners are incentivized by the sender of a transaction to include it into new blocks by receiving fees. While the block reward is determined by the rules of the underlying blockchain, the fees offered can be chosen by the sender of a transaction itself. Nakamoto-style consensus relies on the assumption of rational miners aiming to maximize their own payoff [5]. Naturally, it is more profitable for payoff-maximizing miners to include transactions offering higher fees into new blocks. In fact, the miners can choose arbitrarily which transactions are included into new blocks without violating the rules of the underlying blockchain system [6]. This implies that there is no guarantee for a published transaction to be included into the ledger within a fixed time. In particular, transactions offering low fees may get stuck in the pool of published but unconfirmed transactions for a while [20]. Nevertheless, there are several applications and protocols in which the security of the protocol relies on the parties' ability to publish a transaction on the ledger within a certain time window [1], [8], [9], [16], [18]. Delaying the confirmation of a published transaction may cause financial damage to an honestly behaving party and may lead to successful frauds.

While it is widely assumed that miners participating in the consensus protocol honestly include the most valuable transactions, the assumption that this strategy is most profitable for each miner has been criticized. Eyal and Sirer proposed the selfish mining strategy and showed that it is more profitable for a miner controlling at least 25% of the overall mining power to withhold a mined block [10]. Moreover, rational miners could accept out-of-band bribes to change their mining strategy. This, however, would require an a-priori trust relation between briber and bribee which leads to the introduction of *in-band bribery* in which a conditional bribery is implemented on the blockchain [4]. As the bribery condition is validated on the blockchain, the trust in the ledger replaces the trust assumption between briber and bribee. Based on that, Liao and Katz proposed the concept of "whale" transactions which offer exceptionally high fees to pay in-band bribes to miners in order to incentivize forks leading to a potential ledger history revision attack [12]. Moreover, McCorry et al. [14] investigated the potential of smart contracts for in-band bribery, proposing a smart contract that incentivizes other miners to change their mining strategy such that a malicious miner with at least 25% of the mining power obtains full control over the ledger. Finally, Dong et al. [7] use smart contracts to design a mechanism that addresses the collusion problem in redundant verifiable computation using a game-theoretic mechanism.

The main contribution of our work is to study a much more fine-grained bribery attack. Instead of taking control over the ledger, our attacker aims to perform a temporary censorship against a specific ledger account which may be a smart contract. In our work we assume the attackers mining power to be negligible, such that the attacker cannot mine any blocks on his own. Thus, instead of following a malicious

This work was partly supported by the German Research Foundation (DFG) CRC 1119 CROSSING (project S7) and the German Federal Ministry of Education and Research (BMBF) iBlockchain project.

mining strategy himself, the attacker establishes a mechanism using smart contracts that incentivizes the rational miners to perform the temporary censorship. In particular, we analyze the mechanism as a *Markov game* played by the miners, where each round of the game represents an extension of the ledger. Depending on the miners strategies the game may reach a state that denotes a successful censorship attack, where the mechanism especially incentivizes strategies leading to the states of a successful censorship. If it is not profitable for any miner to deviate from his current strategy, the played strategy profile is stable and commonly referred to as *equilibrium*. In this work, we present three different in-band bribery mechanisms for a temporary censorship attack. We analyze and compare the resulting mechanisms with respect to the success of the censorship attack, the established equilibrium concept and the attack costs for the bribery contract execution.

- The first mechanism pays to a list of miners a fixed bribe if and only if the temporary censorship was successful. We show that the established game has an *equilibrium* for the desired outcome only for bribery costs growing exponentially in the attack time.
- In the second mechanism, compensation are payed for each block mined according to the censorship, instead of rewarding the outcome of the game. Due to this modification, an even stronger *equilibrium* concept is established for linear growing bribery costs, but with the trade-off of also linearly growing on-chain communication overhead.
- Finally, the third mechanism offers a bribe to just a single committing miner. To this end, we revisit the concept of *feather forks* introduced by Miller [15]. We show that the same *equilibrium* as in the previous game can be established for a constant amount of on-chain communication and further reduce the costs if some miner commits to the briber contract before the censorship time.

In Section II, we introduce the fundamentals for our work used in Section III to formalize a general censorship game played by the miners. In Sections III-B to III-C we present and analyze different bribery mechanisms using smart contracts leading to fundamentally different games. Moreover, we discuss aspects of our model and relevance of the bribery attack in Section IV and finally conclude our results in Section V.

II. Background

A. Blockchain

The public ledger L over a set of accounts $\mathcal{A}$ at time t is defined as a set of ordered blocks $\{B_1, ..., B_t\}$. To define a single block, we follow the definition of the Ethereum Yellow paper [21], but for simplicity reduce the definition to the properties required for our work. Each block B is a tuple $B = \{\texttt{num}, \texttt{beneficiary}, \texttt{stateRoot}\}$ where $B.\texttt{num} \in \mathbb{N}$ is the number of the block B, $B.\texttt{beneficiary} \in \mathcal{M}$ is the account address that receives the block reward and the fees for this block and $B.\texttt{stateRoot} \in \{0,1\}^l$ is a hash value that encodes the current world state $\sigma^{(L)}_{\texttt{num}}$. For simplicity, we refer to the block B with $B.\texttt{num} = t$ as B_t. The current world state $\sigma_t^{(L)} = \{\sigma_t^{(a_j)}\}, \forall a_j \in \mathcal{A}$ is a set of all current account states. In Ethereum, the world state is encoded as a root hash of a Merkle Patricia Trie that contains hashes of all account states. For further details we refer the reader to [21]. For the purpose of our work, we assume a cryptographic hash function $H : \{0,1\}^* \mapsto \{0,1\}^l$ that is used to construct a hash tree and a function $checkMerkleProof(h_{root}, P_x, x) \mapsto \{true, false\}$ that verifies if an element x is contained in a hash tree with the root hash h_{root} given a proof P_x. This proof P_x is of size $O(log(n))$, where n is the number of elements in the tree and can be generated and verified efficiently by every party knowing the corresponding elements of the hash tree. For our purpose we conclude that, given a block B_t and an account state $\sigma_t^{(a_j)}$, it is efficiently possible to proof that $\sigma_t^{(a_j)}$ is part of the world state in block B_t and further, given a proof $P_{\sigma_t^{(a_j)}}$, it is efficiently possible to verify if $\sigma_t^{(a_j)}$ is part of the world state at block B_t. The public ledger L is maintained by a set of n miners $\mathcal{M} = \{M_1, ..., M_n\}$. For the block creation we adapt the model used by Liao and Katz [12]. Each miner M_i controls a fraction p_i of the overall computational power, with $\sum_{i=1}^{n} p_i = 1$. The distribution of the computational power amongst the miners remains constant over time and is publicly known, i.e. at each point in time each miner knows the fraction p_i that miner M_i controls $\forall M_i \in \mathcal{M}$. We assume miners generate blocks according to a Poisson process with a constant rate, i.e. time is modeled by block creation events. At time t each miner M_i has a chance p_i to generate a new block B_{t+1}.

A new block B_{t+1} rewards the miner of this block $B_{t+1}.\texttt{beneficiary}$ if the block does not get orphaned over time. The reward is determined by the rules of the underlying blockchain and might depend on the time and previously published blocks [21]. For simplicity we assume that each block B_i rewards the miner with a constant payoff $r \in \mathbb{R}_{>0}$. This reward r covers the determined block reward as well as the expected average fees. This model implies that there are always sufficient unconfirmed transactions of nearly constant fees for the miners to choose from. Moreover, if any transaction Tx offers fees that exceed the average fees by $f \in \mathbb{R}_{>0}$, a miner M_i could increase his payoff to $r + f$ by including Tx into the next new block. We generally assume that any exceeding fees f are significantly lower than the block reward r, such that exceeding fees do not incentivize forking the longest chain. We refer the reader to the work by Liao and Katz about incentivizing forks by whale transactions [12] and assume for our work that $0 \leq f \ll r$ for all exceeding fees.

B. Smart Contracts

In this work, we adapt the formalization of smart contracts modeled by Dziembowski et al. [8]. Informally, a smart contract can be seen as an account on the public ledger that can accept coins and inputs from other accounts and redistribute these coins according to some well-defined rules. We define a *contract type* C over a set of ledger accounts $\mathcal{A}$ as a tuple $C = \{\Lambda, \texttt{fun}_1, \texttt{fun}_2, ...\}$, where $\texttt{fun}_i$ is a function of the contract and Λ is the set of possible contract

storages. Informally speaking, a contract type can be seen as contract code. In this work we define the *contract type* for each of our contracts by showing pseudo code for each function $\texttt{fun}_i$ and defining Λ as the set of storage variables of the contract C. A concrete *contract instance* $\texttt{C}$ on the ledger is defined as tuple $\texttt{C} = \{\sigma, C\}$, where C is the *contract type* and $\sigma \in \Lambda \cup \perp$ is the current state of the *contract instance*. A ledger account $\texttt{Ac} \in \mathcal{A}$ can call function $\texttt{fun}_i$ of a *contract instance* $\texttt{C}$ by sending a transaction to a contract instance $\texttt{C}$ including some coins x and some parameters z. By mining the transaction, the miners execute function $\texttt{fun}_i$ on the contract storage $\texttt{C}.\sigma$ using coins x and parameters z. The execution of a contract function can be seen as a state transformation. Each contract evaluation is assumed to be atomic, by which we mean that it is either executed completely or not at all. Additionally, each contract evaluation can operate at time t on some public parameters provided by the ledger execution environment. Following the execution model of the Ethereum Virtual Machine, these public parameters include the current block number and the list of the previous block hashes $(h_{B_1}, ..., h_{B_{t-1}})$ where $h_{B_i} = hash(B_i)$[1] [21]. Note that every time a rational miner decides to evaluate a contract on a party's transaction, we can assume that it is evaluated correctly such that the public ledger can be trusted w.r.t. correctness.

C. Game-theoretic Concepts

Mechanism design takes an engineering approach to incentivize desired outcomes of game-theoretical models in the presence of rational players. It is assumed that each player follows an individual strategy but assuming self-interested behavior, where the real strategies are generally not known. Additionally, it is assumed that players aim at maximizing their individual payoffs. We use the definition of a *Markov game* which is a tuple (Q, N, A, P, u) [19], where

- Q is a finite set of game states,
- N is a finite set of n players,
- $A = A_1 \times \cdots \times A_n$ is the set of action profiles, where A_i is a finite set of actions available to player i,
- $P : Q \times A \times Q \mapsto [0, 1]$ is the transition probability function, such that $P(q, a, \tilde{q})$ is the probability of transitioning from state q to state $\tilde{q}$ after action profile $a \in A$
- $u = u_1, ..., u_n$ is the set of reward functions, where $u_i : Q \times A \mapsto \mathbb{R}$ is a real-valued payoff function for player i, where $r_i(q, a)$ is the utility a player i expects in state q if action profile a is played.

Let $h_T = (q^0, a^0, q^1, a^1, ..., a^{T-1}, q^T)$ denote a history of a *Markov game* at stage T. Then a *behavioral strategy* $s_i(h_T, a_j)$ for player i returns the probability of playing action $a_j \in A_i$ for history h_T. A *Markov strategy* is a restricted *behavioral strategy* s_i, thus $s_i(h_T, a_j) = s_i(h'_T, a_j)$ if $q_T = q'_T$, where q_T and q'_T are the latest states of h_T and h'_T, respectively. Intuitively, for a *Markov strategy* the played actions depend only on the current state and not the entire history of the game [19]. Therefore, we denote a *Markov strategy* as $s_i(q_t, a_j)$. For strategy profile $s = \{s_1, ..., s_n\}$ and a fixed stage T, given an initial state q_0 we can compute a players *T-stage cumulative expected payoff* as $EU_i^T(q_0, s) = \mathtt{E}_{q_0,s}\big[\sum_{t=1}^{T} u_i(q_t, a_t)\big]$. The strategy profile s can also be denoted as tuple (s_i, s_{-i}) for any player i, where s_i is the individual strategy of player i and s_{-i} are the strategies of all other players. Let now $s = (s_1, ..., s_n)$ be a strategy profile in a *Markov game* (Q, N, A, P, u). We call s an *equilibrium* if, given that all other players $-i$ stick to their strategies s_{-i}, there is no player i who can increase his own utility by changing his strategy to $s'_i \neq s_i$. Formally, s is a *T-stage cumulative expected payoff equilibrium* for a *Markov game* (Q, N, A, P, u) of T stages for initial state q_0 if

$$EU_i^T(q_0, (s_i, s_{-i})) \geq EU_i^T(q_0, (s'_i, s_{-i}))$$

$\forall i \in N, \forall s'_i \in S_i | s'_i \neq s_i$. Further, we call a strategy s_i *dominant* if it is generally at least as profitable for a player i as any other strategy s'_i independent of the other players' strategies s_{-i}. If there exists a strategy profile s that consist of *dominant* strategies for each player, we call this profile *dominant strategy equilibrium*. Formally, for a *Markov game* (Q, N, A, P, u) of T stages for initial state q_0 we call $s = (s_1, ..., s_n)$ a *t-stage cumulative expected payoff dominant strategy equilibrium* if

$$EU_i^T(q_0, (s_i, s_{-i})) \geq EU_i^T(q_0, (s'_i, s_{-i})),$$

$\forall i \in N, \forall s'_i \in S_i | s'_i \neq s_i, \forall s_{-i} \in S_{-i}$. where S_{-i} is the set of all possible strategy profiles of other players. Further, if s is an *equilibrium* in a *Markov game* (Q, N, A, P, u) and consists only of *Markov strategies* for any stages T then we call s a *Markov perfect equilibirum* if it is an *equilibrium* regardless of the starting state of the game [19]. Intuitively, this means that action profiles played by s form an *equilibrium* for every sub-stage independent of following or previous states.

D. Rational Behavior

In this work we consider a public ledger maintained by a set of n rational miners $\mathcal{M} = \{M_1, ..., M_n\}$. We assume that each rational miner M_i individually tries to maximize his own utility as a player in a *Markov game*. When choosing a rational mining strategy, the decision of miner M_i relies only on *common knowledge* $\texttt{ck}$ [3] which includes all account states and all contract instances. Since the public ledger is publicly available for every party and is extended only via the consensus protocol, this assumption is suitable [11]. As we assume instant message propagation, it follows that each published but unmined transaction is also part of $\texttt{ck}$. Further, the set of miners $\mathcal{M}$ and the distribution of mining power $\pi = \{p_1, ..., p_n\}$ is also part of the common knowledge $\texttt{ck}$ as this can be derived from the public ledger [2]. Further, rationality implies that no rational miner chooses a strategy that is dominated by any better strategy. For the block creation process we assume a mining power distribution $\pi = \{p_1, ..., p_n\}$ such that it is generally not profitable for

[1]Note that in Ethereum the execution environment restricts the access the 256 latest block hashes. If the contract requires earlier hash values, the contract could provide a function to store intermediary hash values. This function must be called manually in time as long the required hashes are available

any miner M_i to fork the longest chain or to withhold blocks without further incentivation. To the best of our knowledge this means $p_i < 0.25$ for each miner $M_i \in \mathcal{M}$ [10], [14].

Finally, we do not assume any other rational behaving parties to take part in the games established by our mechanisms. In particular this means that neither the attacker who establishes the bribery mechanism nor the victim of the censorship attack behave rationally. On the one hand, the attacker is assumed to be malicious and therefore should generally not be limited to rational behavior. In fact, the attacker might behave not rational for the sake of hidden external utilities or not rational at all accepting low or negative utilities. On the other hand, it might seem rational for the victim of a censorship attack to try to avoid the censorship attack. In particular, the victim might establish a bribery mechanism to prevent the execution of the bribery contracts used by the attacker, starting an arms race. Although the victim might actually win this arms race against a financially limited attacker, we highlight that this kind of active defensive measure is undesired in the context of decentralized autonomous organizations and off-chain protocols as it obligates additional responsibilities and financial burden to an honestly behaving party. Therefore we assume that the victim behaves honestly according to the contract it tries to execute and publishes his transactions to the best of his knowledge with some exceeding fees f and does not participate in the censorship game.

III. Temporary Censorship Attack

In this section we introduce our temporary censorship attack against a target account $\texttt{Ac}$ in the presence of rational miners. We consider a distributed ledger maintained by a set of n rational miners $\mathcal{M} = \{M_1, ..., M_n\}$ with mining power distribution $\pi = \{p_1, ..., p_n\}$ where each miner M_i controls a fraction p_i. As the attacker $\texttt{A}$ itself does not obtain any mining power, he establishes a bribery mechanism incentivizing the rational miners to perform the censorship. To censor an account $\texttt{Ac}$, the attacker $\texttt{A}$ tries to prevent state changes of this account within a censorship interval $\texttt{CI} = \{B_m, ..., B_{m+t}\}$ of length t. Therefore, we assume the cumulative exceeding fees offered by all transactions changing the state $\sigma_m^{(\texttt{Ac})}$ of the target account $\texttt{Ac}$ in a block B_{m+i} do not exceed f_i. For simplicity, the exceeding fees f_i can only be received in block B_{m+i}. Further, we assume that the expected exceeding fees $(f_1, ..., f_t)$ for the censorship interval are *common knowledge*.

The censorship attack is successful if $\sigma_{m+t}^{(\texttt{Ac})} = \sigma_m^{(\texttt{Ac})}$. Informally, this means that the state of the target account at the begin of the censorship interval equals the state of the target account at the end of the censorship interval. As state changes are assumed to be irreversible it can be concluded that no transaction from or to account $\texttt{Ac}$ has been included.

We can now define the censorship game as a *Markov game* (Q, N, A, P, u) played by the miners $N = \mathcal{M}$ for t stages. The set of states $Q = \{q_0, q_1, q'_1, ..., q_t, q'_t\}$ denotes the possible states of the censorship game for a censorship interval of length t. Informally, each state q_j denotes a state in which a successful censorship is still possible and q'_j denotes a state

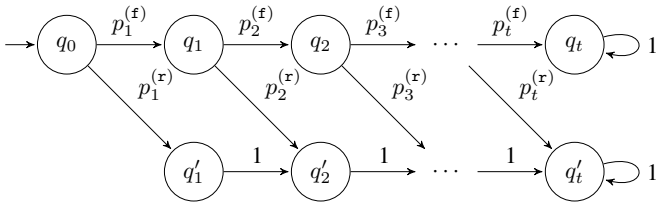

Fig. 1. The state transition function P with edges showing transition probabilities.

where this is impossible, respectively. Therefore, we denote $Q' := \{q'_1, ..., q'_t\}$. The censorship attack is successful if the game ends up in state q_t after t stages. In the initial state q_0 a successful attack is generally assumed to be possible as the attacker aims to prevent state changes of the target account and not to revert history. The set of action profiles $A = A_1 \times \cdots \times A_n$ consists of the set of actions $A_i = \{\texttt{refuse}, \texttt{follow}\}$ for each miner $M_i \in \mathcal{M}$. A miner M_i playing action $\texttt{refuse}$ at some time j means that M_i decided to refuse the censorship and tries to change the state $\sigma_m^{(\texttt{Ac})}$ for potentially receiving the exceeding fees f_j. On the other hand, playing action $\texttt{follow}$ means that the miner follows the censorship and tries to mine a block that does not change the state. If all miners M_i play the same action a_i with $a_i = a_j$ for any $M_j, M_i \in \mathcal{M}$ at any time j, the action profile is denoted as $a^{(a_i)}$. Each state transition in the censorship game denotes a successful extension of the blockchain. The state transition probability function $P : Q \times A \times Q :\mapsto [1, 0]$ is depicted in Figure 1, where $p_j^{(\texttt{f})}$ and $p_j^{(\texttt{r})}$ denote the cumulative hashpower of all miners playing $\texttt{follow}$ and $\texttt{refuse}$, respectively. As state transitions are assumed to be irreversible $P(q'_j, a*, a'_{j+1}) = 1$ for any stage j and any action profile $a* \in A$.

All rational miners are assumed to maximize the expected payoff in the censorship game. To perform the attack, attacker $\texttt{A}$ establishes a bribery mechanism that manipulates the utility function u of the game. Once the attacker publishes a bribery contract, it becomes part of *common knowledge* $\texttt{ck}$. All game parameters (Q, N, A, P, u) as well as the initial state q_0 of the game are known and each miner $M_i \in \mathcal{M}$ may choose a mining strategy. The miners concurrently mine block B_{m+1} of the censorship interval in the first stage of the game according to the played action and append it to the public ledger. After that, the miners individually evaluate their strategy for the next stage and continue mining the next block according to the new action. Assuming that $\texttt{ck}$ initially does not contain any information that incentivizes following the censorship and that the attacker did not establish any bribery mechanism, each miner M_i is incentivized by the exceeding fees. Each miner M_i expects a payoff of $(r + f_j) \cdot p_i$ for each block $B_{m+j} \in \texttt{CI}$ for playing action $\texttt{refuse}$. As no miner is incentivized to fork the blockchain this expected reward is independent of the other players' actions. We thus estimate the expected payoff for miner M_i choosing a *Markov strategy* $s_i^{(\texttt{r})}$ always playing action $\texttt{refuse}$ in the game in the absence of any bribery

contract as

$$EU_i^t(q_0, s^{(r)}) = \sum_{j=1}^{t} (r + f_j) \cdot p_i, \tag{1}$$

Playing a strategy involving action `follow` would yield a lower utility since the miner would at least partially give up some exceeding fees f_j without further reward. Thus, without any bribery contract by the attacker, always playing `refuse` is the dominant strategy for all miners $M_i \in \mathcal{M}$ and the strategy profile $s^{(\mathtt{r})} = \{s_1^{(\mathtt{r})}, ..., s_n^{(\mathtt{r})}\}$ is a *dominant strategy Markov perfect equilibrium*, implying that the censorship is not successful.

In the following we present three different bribery contracts that establish different bribery mechanisms to break the *equilibrium* for $s^{(\mathtt{r})}$. In the analysis, each on-chain interaction is taken into account to calculate the attack costs. Therefore, we assume that each bribery contract interaction by the miners is compensated by the attackers' contract to avoid hidden fee costs that would impact the utility of the rational miners.

A. Pay per Miner

In the first mechanism, the attacker publishes a bribery contract of type `SimpleBribery` that incentivizes the miners to follow the censorship by paying a bribe to all miners if and only if the censorship attack is successful. A contract of this type consists of storage variables $\{\mathtt{Ac}, \sigma_m^{(\mathtt{Ac})}, t, \mathcal{B}, \mathcal{M}\}$, where

- `Ac` is the address of the target account,
- $\sigma_m^{(\mathtt{Ac})}$ is the state of the target account at time m,
- t is the length of the censorship interval,
- $\mathcal{B} = \{b_1, ..., b_n\}$ is the set of bribes, and
- $\mathcal{M} = \{M_1, ..., M_n\}$ is the set of miners.

The `SimpleBribery` contract consist of three functions $\{init, fulfill, refund\}$, where

- $init$ is called on creation at time m and initializes the storage variables. It is up to the miners to check if $\sigma_m^{(\mathtt{Ac})}$ actually equals the current state of account `Ac`.
- $fulfill$ can be called by any party after the censorship interval. On receiving a Merkle proof $P_{\sigma_m^{(\mathtt{Ac})}}$, the contract checks if $\sigma_m^{(\mathtt{Ac})}$ is part of the state in block B_{m+t}. If the check holds, the contract pays the bribe b_i to the miner M_i for each $M_i \in \mathcal{M}$ and terminates.
- $refund$ can be called by the attacker after some timeout Δ at time $m + t + \Delta$ to refund all unclaimed bribes.

In case of a successful censorship attack any miner M_i is able to provide a valid proof to fulfill the ledger contract and claim the bribe. Otherwise, in case of a not successful attack, no miner is able to claim bribes from the contract without hurting the correctness of the ledger. We conclude that the bribes are only paid if the censorship is successful.

To incentivize each miner to perform the censorship, the attacker chooses each bribe b_i such that the utility for a successful censorship attack is higher than the utility for an unsuccessful attack. Let $s = (s_1, ..., s_n)$ be a strategy profile of *Markov strategies* $s_i(\tilde{q}, \mathtt{follow}) = 1$ for each $\tilde{q} \in Q$ for each miner $M_i \in \mathcal{M}$.

Lemma 1. *In a* censorship game (Q, N, A, P, u) *of* t *stages established by contract* `SimpleBribery`*, the strategy profile* s *is a* t-stage cumulative expected payoff equilibrium *if*

$$b_i > \sum_{j=1}^{t} f_j \cdot p_i, \quad \forall M_i \in \mathcal{M} \tag{2}$$

On the other hand let $\tilde{s} = (\tilde{s}_1, ..., \tilde{s}_n)$ be a strategy profile, where each miner M_i chooses strategy $\tilde{s}_i(\tilde{q}, \mathtt{refuse}) = 1$, consequently playing `refuse`.

Lemma 2. *In a* censorship game (Q, N, A, P, u) *of* t *stages established by contract* `SimpleBribery`*, the strategy profile* $\tilde{s}$ *is a* t-stage cumulative expected payoff equilibrium *if*

$$b_i \leq \frac{\sum_{j=1}^{t} (f_j \cdot p_i^j)}{p_i^t} \tag{3}$$

for all miners $M_i \in \mathcal{M}$.

Finally, let $\hat{s} = (\hat{s}_1, ..., \hat{s}_n)$ be a strategy profile where each miner M_i chooses a *Markov strategy* $\hat{s}_i$ with

$$\begin{aligned} \hat{s}_i(\tilde{q}_j, \mathtt{follow}) &= \begin{cases} 1, & \mathtt{if}\ \ \tilde{q}_j \in Q \setminus Q' \\ 0, & \mathtt{otherwise} \end{cases} \\ \hat{s}_i(\tilde{q}_j, \mathtt{refuse}) &= \begin{cases} 1, & \mathtt{if}\ \ \tilde{q}_j \in Q' \\ 0, & \mathtt{otherwise}. \end{cases} \end{aligned} \tag{4}$$

Intuitively this means each miner plays `follow` as long as it is possible to achieve a successful censorship and `refuse` otherwise.

Lemma 3. *In a* censorship game (Q, N, A, P, u) *of* t *stages established by contract* `SimpleBribery`*, the strategy profile* $\hat{s}$ *is a* dominant strategy t-stage cumulative expected payoff equilibrium *if*

$$b_i > \frac{\sum_{j=1}^{t} (f_j \cdot p_i^j)}{p_i^t} \tag{5}$$

for all miners $M_i \in \mathcal{M}$.

We argue that, if inequation 3 holds for every miner M_i, the censorship game contains two different *equilibria*. Thus, the game does not guarantee the success of the censorship attack, although the *equilibrium* leading to a successful attack might be individually more profitable for each miner M_i. In case of an unsuccessful censorship attack, choosing strategy $\tilde{s}_i$ instead of s_i would have been more profitable for each miner M_i. Reaching the *equilibrium* s thus requires additional coordination between the miners as the miners do not know each other's strategy. Only the dominance of strategy profile $\hat{s}$ leads to a successful censorship attack without further coordination. In this case we can estimate the attack costs

$$\mathcal{C}_{attack} > \mathcal{C}_{\mathtt{SimpleBribery}} + \mathcal{C}_{\mathtt{proof}} + \sum_{i=1}^{n} \frac{\sum_{j=1}^{t} (f_j \cdot p_i^j)}{p_i^t}, \tag{6}$$

where $\mathcal{C}_{\mathtt{SimpleBribery}}$ are deployment costs for the bribery contract and $\mathcal{C}_{\mathtt{proof}}$ are execution cost to prove a successful bribery that the attacker compensates. Note that contract

SimpleBribery implies that the bribe reward is only payed in stage t. It thus follows that $u_i(q_j, a^{(\texttt{refuse})}) > u_i(q_j, a^{(\texttt{follow})})$ $\forall q_j \in Q \setminus \{q_{t-1}\}, \forall M_i \in \mathcal{M}$, i.e. the expected stage reward for playing refuse is generally more profitable as the miner could additionally receive the exceeding fees as stage reward. We conclude that even if $\hat{s}$ is a *cumulative dominant strategy equilibrium*, it is generally not a *Markov perfect equilibrium*.

B. Pay per Block

The second bribery mechanism pays a bribe for each block of the censorship interval mined according to the censorship conditions. To this end, this bribery mechanism offers a compensation for each block of the censorship interval CI as long as the state of the target account $\sigma_m^{(\texttt{Ac})}$ is not changed. Therefore, the attacker publishes a bribery contract of type CompensationBribery at time m for a censorship interval $\texttt{CI} = \{B_m, ..., B_{m+t}\}$. A contract of this type uses the storage variables $\{\texttt{Ac}, \sigma_m^{(\texttt{Ac})}, t, \mathcal{B}\}$, where

- Ac is the address of the target account,
- $\sigma_m^{(\texttt{Ac})}$ is the state of the target account at time m,
- t is the length of the censorship interval, and
- $\mathcal{B} = \{b_1, ..., b_t\}$ is the set of compensation bribes, with each bribe $b_i > f_i \forall i \in [t]$.

Furthermore, a contract of the type CompensationBribery contains three functions $\{init, compensate, refund\}$, where

- $init$ is called on creation at time m and initializes the storage variables. It is up to the miners to check if $\sigma_m^{(\texttt{Ac})}$ actually equals the current state of account Ac.
- $compensate$ can be called by any party after time $m+j$ to claim a compensation for block B_{m+j}. On receiving a Merkle proof $P_{\sigma_m^{(\texttt{Ac})}}$ the contract validates the proof to check if $\sigma_m^{(\texttt{Ac})}$ is part of the state in block B_{m+j}. If the check holds and the compensation for this block has not been paid yet, the contract pays the compensation b_j to the miner $M_i = B_{m+j}.\texttt{beneficiary}$ and internally marks the block B_{m+j} as compensated. If all blocks are marked, the contract terminates.
- $refund$ can be called by the attacker after some timeout Δ at time $m + t + \Delta$ to refund all unclaimed bribes.

Trivially, the censorship attack is successful if the state of the target account Ac remains unchanged during the censorship interval. The censorship attack is successful if it is possible to submit a proof for each block $B_i \in \texttt{CI}$. If the account state $\sigma_m^{(\texttt{Ac})}$ changed at time $m + j$, it is not possible to claim any compensation from the ledger contract for a block B_k with $k \geq m + j$. Let now $\hat{s} = \{\hat{s}_1, ..., \hat{s}_n\}$ be a strategy profile, where each miner M_i plays the strategy $\hat{s}$ defined in 4.

Lemma 4. *In a* censorship game (Q, N, A, P, u) *of* t *stages established by contract* CompensationBribery*, the strategy profile* $\hat{s}$ *is a* dominant strategy Markov perfect equilibrium *if*

$$b_i > f_i, \forall i \in [t] \tag{7}$$

Intuitively, the contract pays b_j as compensation for not playing refuse. As $b_j > f_j$ it is individually more profitable to play action follow in every stage $q_j \in Q \setminus Q'$. Assuming that all rational miners follow their strategy of the *dominant strategy Markov perfect equilibrium*, all miners will play the follow action. Therefore, the game will not reach a state $q' \in Q'$. An induction over the game states implicates that the censorship game guarantees a successful censorship attack if all miners behave rationally. For this kind of bribery contract, we can estimate the attack costs as

$$\mathcal{C}_{attack} > \mathcal{C}_{\texttt{CompensationBribery}} + \mathcal{C}_{\texttt{proof}} \cdot t + \sum_{i=1}^{t} f_i, \tag{8}$$

where $\mathcal{C}_{\texttt{CompensationBribery}}$ are the costs for contract deployment and $c_{\texttt{proof}}$ are the costs for single proof validation by calling the $compensate$ function. Note that for this bribery contract the attack cost grows only linearly in the length t of the censorship interval. However, the on-chain execution costs the attacker A has to compensate also grow linearly in t. Finally, the attacker may at least partially pay for an unsuccessful censorship attack if some miners do not behave rationally.

C. Pay per Commit

In this section we finally present a bribery mechanism that pays just one single miner M_b with mining power p_b to perform a temporary censorship attack against any target account Ac. Therefore, we revisit the concept of *feather forking* introduced by Andrew Miller [15]. In a feather fork scenario, one miner $M_b \in \mathcal{M}$ with mining power p_b wants to perform a censorship attack against an account Ac. The miner M_b publicly announces to fork any block at the head of the blockchain that contains any transaction that changes $\sigma_m^{(\texttt{Ac})}$. As M_b only holds a fraction $p_b \ll 0.5$ of the overall mining power he has only a very small chance of mining an alternative longest chain if all other miners would mine on the original longest chain. Therefore, M_b gives up his fork if the block he wants to fork has at least one confirming child block. Due to the public announcement, the other miners know if any miner M_i publishes a block $\tilde{B_n}$ changing $\sigma_m^{(\texttt{Ac})}$ the miner M_b tries to publish an alternative longest chain with the head B_n and B_{n+1} before $\tilde{B_n}$ gets confirmed by any other block $\tilde{B}_{n+1}$. If M_b is successful, the block $\tilde{B_n}$ gets orphaned as all miners extend the longest chain and the miner M_i would not receive any rewards for block $\tilde{B_n}$. The chance for M_b to create two blocks in a row before any other miner generates one block is p_b^2, where p_b is the chance for miner M_i to generate one block. This means for block $\tilde{B_n}$ that there is a chance of p_b^2 to get orphaned such that it might be more profitable for miner M_i to follow the censorship instead of risking orphaned blocks.

To use this concept for our analysis, it is of the essence that the announcement of some miner M_b is committing such that each miner can be sure that a rationally behaving miner M_b will stick to his announcement. To this end, it must be more profitable for M_b to stick to his announcement than to deviate from it at any stage of the censorship game. To establish this mechanism, the attacker A publishes a bribery contract of type FeatherForkBribery at any time n before

the attack starts. The contracts storage variables are defined as $\{\texttt{Ac}, \sigma_m^{(\texttt{Ac})}, t, m, b, c, M_b\}$, where

- $\texttt{Ac}$ is the address of the target account,
- $\sigma_m^{(\texttt{Ac})}$ is the state of the target account at time m,
- t is the length of the censorship interval,
- m is the begin of the censorship interval,
- b is the briber offered to the announcing miner M_b,
- c is the expected deposit by a announcing miner M_b, and
- M_b is the address of the announcing miner.

Bribery contracts of this type consist of the functions $\{init, commit, fulfill, refund\}$, where

- $init$ is called on creation at time n, initializes all storage variables and sets $M_b = \perp$. It is up to the miners to check if $\sigma_m^{(\texttt{Ac})}$ actually equals the current state of account $\texttt{Ac}$ and that it does not change before time m.
- $commit$ is called by any miner M_i who is willing to commit to a censorship announcement at any time m' with $n \leq m' \leq m$ including c coins as deposit. If the deposit is sufficient, the contract sets $M_b = M_i$.
- $fulfill$ can be called by any miner at any time after the censorship interval if $M_b \neq \perp$. On receiving a Merkle proof $P_{\sigma_m^{(\texttt{Ac})}}$, the contract verifies if $\sigma_m^{(\texttt{Ac})}$ is part of the world state in block B_{m+t}. If the check holds, the contract pays $b+c$ coins to the miner M_b and terminates.
- $refund$ can be called by the attacker after some timeout Δ at time $m+t+\Delta$ or after time m if $M_b = \perp$ to refund b coins. The contract may keep any unclaimed deposit by M_b and terminate.

As we assume instant communication, n may equal m in our model, i.e. contract creation, commitment and begin of the censorship interval will happen in the same block. If no miner calls function $commit()$ before the censorship interval, we can assume the censorship attack fails automatically and the attacker reclaims bribe b. However, once M_b is committed, all other miners know that M_b can achieve the desired outcome by performing a *feather fork* every time a block $\tilde{B}_{m+j}$ that changes the state $\sigma_m^{(\texttt{Ac})}$ is published. The attacker has to choose the commitment deposit c and the bribe b when establishing the mechanism such that it is a dominant strategy for M_b to perform a *feather fork* if necessary.

Lemma 5. *In a* censorship game (Q, N, A, P, u) *of* t *stages established by contract* `FeatherForkBribery`*, strategy profile* $\hat{s}$ *is a* dominant strategy Markov perfect equilibrium *for miner* $M_b \in \mathcal{M}$ *committed with deposit* c *and controlling hashpower* p_b *if*

$$c > \frac{\sum_{j=1}^{t} f_j \cdot p_b}{p_b^2} \tag{9}$$

and

$$f_j < \frac{p_b^2 \cdot r}{1 - p_b^2}, \forall f_j \in \{f_1, ..., f_t\} \tag{10}$$

Finally, we note that it is generally profitable for a miner M_b to commit to the bribery contract if its controlled mining power p_b implies a *dominant strategy Markov perfect equilibrium* in the censorship game and the bribe $b > \sum_{j=1}^{t} f_j \cdot p_b$ exceeds the expected cumulative exceeding fees. We conclude that for this censorship attack the attacker A has to pay

$$\mathcal{C}_{attack} > \mathcal{C}_{\texttt{FeatherForkBribery}} + \mathcal{C}_{commit} + \mathcal{C}_{\texttt{proof}} + \sum_{i=1}^{t} f_i \cdot p_b, \tag{11}$$

for a committing miner M_b controlling $p_b > \left|\sqrt{\frac{f_j}{r+f_j}}\right|$ for every $j \in \{1, ..., t\}$. Note that for this attack the paid bribe grows only linearly in time t while the on-chain costs remain constant. In this game, the dominance of the *equilibrium* $\hat{s}$ depends on the hashpower p_b of the committed miner. However, this censorship attack requires an additional commitment of a single miner before the actual attack starts. Although we showed parameters that lead to *dominant strategy Markov perfect equilibrium* in the censorship game and therefore to a successful censorship attack, the censorship attack fails automatically if no miner commits.

IV. Discussion

The presented bribery mechanisms take place within and are restricted to our specified model. We now discuss the impact and practical relevance of temporary censorship bribery in a wider context and the main aspects of the assumed model.

a) Contract Censorship: In our model we assumed that attacker and miner know the exceeding fees of transactions that may change the state of the target account. While it may be hard to estimate the exceeding fees for the target account controlled by some user, we think it may be much easier for a contract account. In fact, many contracts used in protocols implement state machines that expect just one specific transaction at some state within a specific time [1], [8], [18]. For this kind of contract, a successful temporary censorship attack could lead to a successful attack in the underlying protocol. Therefore, we highlight the relevance of presented temporary censorship attacks for so-called off-chain protocols in contrast to general censorships against specific user accounts. To the best of our knowledge, there is no state channel protocol that covers resistance to censorship attacks.

b) Counterbribery: While we assume *common knowledge rationality* for miners, there might be other rational players that are not represented in our model but may actively try to prevent the bribery. Any player P trying to prevent the censorship by offering higher fees or performing another form of counter bribery may actually succeed. Nevertheless, P would have to invest additional money to prevent the attack which might not be intended if P is generally honest. Moreover, the concept of counterbribery expects the victims to proactively watch for potential bribery. We follow Bonneau and argue that this mitigation is undesirable [4]. In the context of state channel protocols, this would expect protocol participants not just to listen on-chain for protocol-related transactions [13] but also for potential bribery attempts.

c) Rational Miners: Although the security of Nakamoto-style consensus is based on incentives and thus assumes rational mining behavior, there is to the best of our knowledge no evidence that any rational miner attacker ever happened [10], [14], [15]. Rational behavior as assumed in our model expects miners to proactively watch for opportunities to increase their payoff and adapt their mining strategy accordingly. We follow Bonneau in arguing that today's miners might be too simplistic to perform this estimation and thus do not recognize bribery attempts [4]. Nevertheless, miners exhibited rational behavior by mining on different cryptocurrencies that are more profitable in the short term [14]. On the other hand, public blockchain miners create blocks concurrently but share concerns about the long-term stability of the cryptocurrency. So, even if following a bribery might be profitable in the short term, the public awareness of a successful bribery may have a negative long-term effect on the value of the underlying currency. Nevertheless, we believe that long-term effects of successful fine-grained temporary censorship attacks will be less crucial than the effects of more powerful attacks where the attacker gains full control over the ledger [4], [14]. We believe that rational mining behavior should be considered when designing incentive-compatible mechanisms for incentive-driven blockchain technology and applications.

d) Accountability: Generally, it is not in the interest of a briber and a bribee to make their relation public. While it is suitable to assume that a malicious attacker is publicly involved in the bribery, a rational miner that values non-monetary utilities such as reputation might prefer to keep its bribery involvements private. While in the first two presented bribery scenarios the miners could potentially obfuscate their involvement by creating new accounts to publish blocks following the censorship, the bribery mechanism established by the `FeatherForkBribery` contract requires an undeniable commitment of a single miner. Therefore, a rational miner that always avoids undeniable bribery involvements would never commit to a feather fork. However, non-monetary preferences are out of scope and should be part of future work.

V. Conclusion

The impact of a censorship attack against the blockchain hurts the widely assumed availability and might cause crucial financial damage and undesired behavior in contract-based protocols. In this paper, we propose a temporary censorship attack based on in-band bribery contracts. Our work shows that, in the presence of rational miners, an attacker is temporarily able to prevent state changes of a designated target account without own mining power. We do not claim that the presented bribery mechanisms are feasible attacks for today's real world blockchain systems as most miners today follow a simple and well-accepted strategy. We emphasize however that these attacks might become real in the future of enhanced mining agents. We define a censorship game played by the miners and show three different bribery contracts that can be published by the attacker to establish fundamentally different mechanism that incentivizes the outcome of a successful censorship. The contracts offer bribes that may be claimed by miners under defined conditions. While in the simplest mechanism all miners may claim a bribe for a successful censorship, we show that the attacker‘s cost for this mechanism grow exponentially in the attack duration. The second mechanism pays bribes for each block mined according to the censorship. In this case we show the existence of a *dominant strategy* that is even dominant in each sub-stage of the temporary censorship, i.e. miners are incentivized to follow the censorship even without expectations about its success. We show with the third contract that it is even possible to establish the same *dominant strategy equilibrium* for significantly lower on-chain costs if one of the rational miner is willing to commit to the success of the censorship beforehand. Finally, we point out the special meaning of fine grained censorship attacks in the context of off-chain protocols that rely on the security to execute smart contracts in time. The code of the bribery contracts and further details on our results are given in the full version of this paper.

Acknowledgment

We thank Stefan Dziembowski and Patrick McCorry for useful discussions, and the anonymous reviewers for their valuable feedback on this work.

References

[1] Raiden network. Aviable: https://raiden.network/. Accessed: 13 -June -2018.

[2] Blockchain.info: estimation of hasrate distribution. Aviable: https://www.blockchain.com/pools, 2018. Accessed: 09-September-2018.

[3] B Douglas Bernheim. Rationalizable strategic behavior. *Econometrica: Journal of the Econometric Society*, pages 1007–1028, 1984.

[4] Joseph Bonneau. Why buy when you can rent? In *International Conference on Financial Cryptography and Data Security*, pages 19–26. Springer, 2016.

[5] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 104–121. IEEE, 2015.

[6] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 2014.

[7] Changyu Dong, Yilei Wang, Amjad Aldweesh, Patrick McCorry, and Aad van Moorsel. Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 211–227. ACM, 2017.

[8] Stefan Dziembowski, Lisa Eckey, Sebastian Faust, and Daniel Malinowski. Perun: Virtual payment channels over cryptographic currencies. Technical report, IACR Cryptology ePrint Archive, 2017: 635, 2017.

[9] Stefan Dziembowski, Sebastian Faust, and Kristina Hostáková. General state channel networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 949–966. ACM, 2018.

[10] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*, pages 436–454. Springer, 2014.

[11] Joseph Y Halpern and Rafael Pass. A knowledge-based analysis of the blockchain protocol. *arXiv preprint arXiv:1707.08751*, 2017.

[12] Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In *International Conference on Financial Cryptography and Data Security*, pages 264–279. Springer, 2017.

[13] Patrick McCorry, Surya Bakshi, Iddo Bentov, Andrew Miller, and Sarah Meiklejohn. Pisa: Arbitration outsourcing for state channels. *IACR Cryptology ePrint Archive*, 2018:582, 2018.

[14] Patrick McCorry, Alexander Hicks, and Sarah Meiklejohn. Smart contracts for bribing miners.

[15] Andrew Miller. Feather-forks: enforcing a blacklist with sub-50% hash power. Available: https://bitcointalk.org/index.php?topic=312668.0, 2013. Accessed: 12-February-2019.
[16] Andrew Miller, Iddo Bentov, Ranjit Kumaresan, and Patrick McCorry. Sprites: Payment channels that go faster than lightning. *CoRR abs/1702.05812*, 2017.
[17] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
[18] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. *draft version 0.5*, 9:14, 2016.
[19] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems*. 2009.
[20] Ingo Weber, Vincent Gramoli, Alex Ponomarev, Mark Staples, Ralph Holz, An Binh Tran, and Paul Rimba. On availability for blockchain-based systems. In *Reliable Distributed Systems (SRDS), 2017 IEEE 36th Symposium on*, pages 64–73. IEEE, 2017.
[21] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

APPENDIX

A. Equilibria Proofs Part 1

For this proof, consider a *censorship game* (Q, N, A, P, u) of t stages defined by the contract `SimpleBribery` as part of *common knowledge ck*.

1) Proof of Lemma 1: In case of a strategy profile $s = (s_1, ..., s_n)$ where each miner M_i chooses a strategy $s_i(\tilde{q}, \texttt{follow}) = 1$ for each state $\tilde{q} \in Q$, we can compute the expected cumulative utility for t stages as

$$EU_i^t(q_0, (s_i, s_{-i})) = t \cdot p_i \cdot r + b_i \tag{12}$$

for every miner $M_i \in \mathcal{M}$. If, in this case, a single miner M_i decided to choose strategy $s_i' \neq s_i$, this could only reduce the miner's payoff. For each strategy $s_i' \neq s_i$ leading to the same played actions the, expected payoff does not change as the utility function u_i depends on the played actions. Finally, if M_i follows any strategy s_i' that plays the action `refuse` at least one time and M_i would successfully create a block according to the `refuse` action at stage j, he would receive the exceeding fees f_j but it would not be possible to receive the bribe b_i anymore. The miner M_i can not expect to receive exceeding fees f_j without giving up b_i. Therefore, if all other miner follow the strategy profile s_{-i}, it is not profitable for any miner M_i to play any strategy $s_i' \neq s_i$ if

$$b_i > \sum_{j=1}^{t} f_j \cdot p_i, \tag{13}$$

for all miner $M_i \in \mathcal{M}$. Informally, this means that the expected reward for the exceeding fees is always lower than the bribe offered to the miner. We conclude that the strategy profile s is a *t-stage cumulative expected payoff equilibrium* for the *censorship game* (Q, N, A, P, u) of t stages defined by the contract `SimpleBribery` if the condition 13 is satisfied. □

2) Proof of Lemma 2 and Lemma 3: In case of a strategy profile $\tilde{s} = \{\tilde{s}_1, ..., \tilde{s}_n\}$ where each miner M_i chooses a strategy $\tilde{s}_i(\tilde{q}, \texttt{refuse}) = 1$ for each state $\tilde{q} \in Q$ we can compute the expected cumulative utility for t stages as

$$EU_i^t(q_0, (\tilde{s}_i, \tilde{s}_{-i})) = \sum_{j=1}^{t} (r + f_j) \cdot p_i \tag{14}$$

In this case any miner M_i deviating from $\tilde{s}_i$ would reduce his expected payoff by the opportunity of receiving the exceeding fees f_j for every time j at which M_i decides to play the `follow` action. Only the case of a successful censorship attack increases the payoff that miner M_i expects. Therefore, M_i may deviate from strategy $\tilde{s}_i$ by choosing strategy $\hat{s}_i$ instead. Informally, M_i tries to receive the bribe as long as it is possible even if all other miner always play `refuse`. Formally, the expected utility for the deviating miner M_i can be computed as

$$EU_i^t(q_0, (\hat{s}_i, \tilde{s}_{-1})) = EU_i^t(q_0, (\tilde{s}_i, \tilde{s}_{-i})) + p_i^t \cdot b_i - \sum_{j=1}^{t} f_j \cdot p_i, \tag{15}$$

where p_i^t is the probability for miner M_i to mine all blocks in the censorship interval `CI` and receive the bribe b_i, and $\sum_{j=1}^{t}(f_j \cdot p_i^j)$ are the expected exceeding fees that M_i gives up for this opportunity. Note that M_i gives up the exceeding fees f_j only if he mined all the previous blocks $B_{m+1}, ..., B_{m+j-1}$. We conclude that also the strategy profile $\tilde{s}$ is a *t-stage cumulative expected payoff equilibrium* if

$$b_i \leq \frac{\sum_{j=1}^{t}(f_j \cdot p_i^j)}{p_i^t} \tag{16}$$

holds. □

Otherwise, the strategy profile $\tilde{s}$ is not *stable* meaning that a strategy $\hat{s}_i$ may dominate strategy $\tilde{s}_i$ for a subset of miners. But if

$$b_i > \frac{\sum_{j=1}^{t}(f_j \cdot p_i^j)}{p_i^t} \tag{17}$$

for all miners $M_i \in \mathcal{M}$, the strategy profile $\hat{s}$ is a *dominant strategy t-stage cumulative expected payoff equilibrium*. As the state transaction function $P(p_j, a^{(\texttt{follow})}, p_{j+1}) = 1$ for all $p_j \in Q \setminus Q'$, the strategy profile $\hat{a}$ results in the same played actions and state transitions as the profile s and therefore to the same expected payoff. Note that in this case the strategy profile s from Lemma 1 is still an equilibrium. But as the profile $\hat{s}$ maximizes the expected payoff independent of the other miners' strategy $\hat{s}$ dominates s. □

B. Equilibria Proofs Part 2

For this proof, consider a *censorship game* (Q, N, A, P, u) of t stages defined by the contract `CompensationBribery` as part of *common knowledge ck*.

1) Proof of Lemma 4: The contract `CompensationBribery` pays bribes for each block as long as the state of the target account `Ac` is not changed. Once the state is changed, it is not possible to claim any compensation for subsequent blocks. Therefore, in a censorship game defined by a bribery contract `CompensationBribery`, the attacker could expect to receive the compensation only in states $\tilde{q} \in Q \setminus Q'$. Generally, if $b_i > f_i \forall i \in [t]$, the utility function in the censorship game defined by `CompensationBribery` can be estimated by

$$u_i(\tilde{q}, (\texttt{follow}, a^*_{-i})) > u_i(\tilde{q}, (\texttt{refuse}, a^*_{-i})) \tag{18}$$

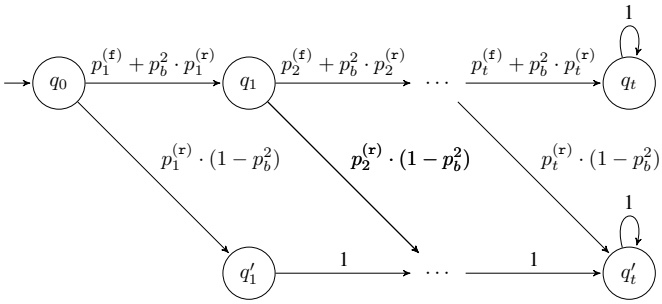

Fig. 2. The state transition function P, where each edge shows the transition probability in case of one miner M_b controlling hash power p_b committed to the feather fork censorship.

$\forall \tilde{q} \in Q \setminus Q'$ and

$$u_i(\tilde{q}, (\texttt{follow}, a^*_{-i})) < u_i(\tilde{q}, (\texttt{refuse}, a^*_{-i})) \tag{19}$$

$\forall \tilde{q} \in Q'$ respectively for all miners $M_i \in \mathcal{M}$ and for any action profile a^*_{-i} played by the other miners. Therefore, for any miner M_i, the expected stage reward at any stage j depends only on the current state $\tilde{q}_j$ and the action played by the miner a_i. From 18 and 19 it trivially follows that playing `follow` dominates playing `refuse` in every state $\tilde{q} \in Q \setminus Q'$ and vice versa for $\tilde{q} \in Q'$ in any stage of the game for every miner $M_i \in \mathcal{M}$. We conclude for the censorship game (Q, N, A, P, u) defined by the bribery contract `CompensationBribery` that the *Markov strategy* $\hat{s}_i$ as defined in 4 is a dominant strategy and the strategy profile $\hat{s}$ is a *dominant strategy expected payoff Markov perfect equilibrium.* □

C. Equilibria Proofs Part 3

For this proof, consider a *censorship game* (Q, N, A, P, u) of t stages defined by the contract `FeatherForkBribery` as part of *common knowledge ck*.

1) Proof of Lemma 5: This proof consist of two different parts. First we show that following the censorship dominates refusing it if one miner M_b strictly follows a feather forking strategy. Then we show that a rational miner will strictly follow if the initial funds committed to the contract are sufficient.

Due to feather forking, there is a chance that a block that changes the state of the target account `Ac` gets orphaned over time. For simplicity, we assume that, if a miner M_b controlling hashpower p_b is committed to the contract, every time the game state transitions from state $q_j \in Q \setminus Q'$ to state $q'_{j+1} \in Q'$, there is a chance of p_b^2 transitioning to state q_{j+1} instead as depicted in Figure 2.

Assuming that a miner M_b is committed to the contract `FeatherForkBribery`, the utility function u_i at state $\tilde{q}_j \in Q \setminus Q'$ can be estimated for each miner $M_i \in \mathcal{M} \setminus \{M_b\}$ as

$$u_i(\tilde{q}_j, (a_i, a^{(b)}_{-i})) \begin{cases} = r \cdot p_i & : a_i = \texttt{follow} \\ \leq (r + f_j) \cdot p_i \cdot (1 - p_b^2) & : a_i = \texttt{refuse} \end{cases} \tag{20}$$

where $a^{(b)}_{-i}$ is the action profile played by the other miners with $a_b = \texttt{follow}$ as M_b is committed to this action. Note that this estimation assumes that the committed miner tries at least to perform a *feather fork* if some block according to action `refuse` is published.[2] We can conclude that for a miner M_i playing $a_i = \texttt{follow}$ dominates playing $a_i = \texttt{refuse}$ for all states $\tilde{q}_j \in Q \setminus Q'$ if $u_i(\tilde{q}_j, (\texttt{follow}, a^{(b)}_{-i})) > u_i(\tilde{q}_j, (\texttt{refuse}, a^{(b)}_{-i})$ for all $f_j \in \{f_1, ..., f_t\}$. Following the equation from 20, this is true if

$$f_j < \frac{p_b^2 \cdot r}{1 - p_b^2} \tag{21}$$

for all $f_j \in \{f_1, ..., f_t\}$. For all states $\tilde{q}_j \in Q'$, we assume for simplicity that playing action $a_i = \texttt{refuse}$ dominates playing $a_i = \texttt{follow}$ as the miner M_b would not continue trying to fork once he failed to perform a *feather fork*. Therefore we conclude that strategy $\hat{s}_i$ is dominant for every miner $M_i \in \mathcal{M} \setminus \{M_b\}$ at any stage of the game.

For the committed miner M_b, playing action $a_b = \texttt{follow}$ means at least trying to mine a block that does not change $\sigma_m^{(\mathtt{Ac})}$ and trying to perform a *feather fork* if someone published a block that does. Playing action $a_b = \texttt{refuse}$ means not even trying to perform a *feather fork* and instead trying to receive the exceeding fees f_j. A miner M_b that plays $a_b = \texttt{refuse}$ at any stage of the game risks to lose c. Therefore, we choose c such that, once M_b is committed, it is generally not profitable to play action $a_b = \texttt{refuse}$ in any state $\tilde{q}_j \in Q \setminus Q'$.

We argue that in every state q_{j-1}, if a block B_{m+j} changing $\sigma_m^{(\mathtt{Ac})}$ is published, M_b has an opportunity to perform a successful *feather fork* with probability p_b^2. If M_b succeeds, he retains c and the censorship game transforms to state q_j, otherwise he could accept the block instead by charging off c and the game state transforms to q'_j. In the latter case, it is generally profitable for M_b to play $a_b = \texttt{refuse}$ in all subsequent states, trying to receive exceeding fees $(f_{j+1}, ..., f_t)$. We conclude that the expected utility for retaining c should always exceed the expected cumulative exceeding fees independent of the block rewards, thus we estimate c as

$$c > \frac{\sum_{j=1}^{t} f_j \cdot p_b}{p_b^2}. \tag{22}$$

Therefore we conclude that $\hat{s}_b$ playing `follow` as long as it is possible is also a dominant strategy for M_b if he is committed by value c. In this censorship game, the strategy profile $\hat{s}$ is a *expected payoff dominant strategy Markov perfect equilibrium*. □

[2]For simplicity we omit any other actions that the committed miner may take and use his probability of successful feather forking as lower boundary.