´

Selfish Mining and Networking Effects

Thesis zur Erlangung des Grades
Master of Science (M. Sc.)
im Studiengang Computer Science

Alexander Wagner
wagner.2@campus.tu-berlin.de

Distributed Security Infrastructures
Institut für Softwaretechnik und Theoretische Informatik
Fakultät Elektrotechnik und Informatik
Technische Universität Berlin

Gutachter:
Prof. Dr. Florian Tschorsch
TBA: Second supervisor

eingereicht am: TBA

Hiermit erkläre ich an Eides statt, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein.

Berlin, TBA

.....

(Alexander Wagner)

Zusammenfassung

Abstract

Contents

Chapter 1

Introduction

Bitcoin is the most prominent example of a decentralized cryptocurrency [2]. Before the development of Bitcoin a decentralized cryptocurrency had been envisioned for many years. It is a system, where a ledger is kept consistent among multiple parties in a peer-to-peer network without the need of trust. It enables the deployment of electronic cash without a central authority figure like a bank. For this reason it is an enhancement to the currently established electronic banking system.
A consistent distributed ledger is essentially a consensus problem, which has to be solved in a cooperative, distributed manner. It is therefore a Byzantine Agreement problem [5]. Bitcoin assumes an honest majority in a public system [13]. Thus, the consistence and correctness of the ledger reduces to a voting problem. However, voting in a public distributed system remains a hard problem, especially considering sybil attacks [3]. Bitcoin reduces the effectiveness of sybil attacks by binding voting right to computational power. In order for a peer to participate in the system, he has to solve a cryptographic puzzle. This process, also known as mining, consumes the computational ressources of the peer. Since there would be no reason to waste computational ressources without gain, mining is incentivized. A miner receives a so called mining reward for mining a block. This incentivized process helps spreading the overall computational power of the network among multiple different parties, since every party is competing for mining rewards. Since mining is inherently constructed through incentives, miners will strive for the best strategy to maximize rewards. Eyal and Sirer show the existence of deviant mining protocols with greater rewards. Miners executing such protocols are called selfish miners. This imposes a threat, since it reduces the performance of the overall system. Additionally selfish miners obtain a greater voting power than their computational ressources allow and as a result tilt the honest majority balance.
The central goal of this master thesis is to analyze the impact of selfish mining as an attack on blockchain systems. While it has been established that selfish mining imposes a threat on blockchain, it remains unassessed how big the impact is. Additionally, selfish mining is highly influenced by networking effects. Therefore, in order to assess the impact of selfish mining, analysis has to be performed in a model, which also captures the underlying network.

Chapter 2

Related Work

Selfish mining is a statistical attack. To analyze profitibality it is therefore beneficial to analytically model selfish mining. In order to study the impact of deviating mining strategies it is very important to represent the blockchain network as close to reality as possible in a mining model, to estimate realistic results. In the following recent selfish mining models as well as network models will be discussed.

2.1 Selfish Mining Models

Blockchain Mining is most commonly modelled through markov decision processes. It is generally used to model decision making, where the outcomes are influenced by random processes and the decision of the decision maker [8]. For the case of selfish mining the selfish miner chooses his next action, so he controls the decision making process. The rest of the network, the block arrival and block propagation can be modelled by stochastic processes.
Utilizing a markovian model revenue gains can be analytically estimated. Eyal and Sirer first described a selfish mining model. Eyal and Sirer used a Monte Carlo simulation to generate blocks and 1000 Miners operating at identical rates. Block propagation time is considered negligible compared to block generation time [4]. Therefore, communication is considered to be instantaneous. In the case of two branches of identical length, the miners are split up into factions mining on one of the two branches based on the network factor γ [4]. γ resembles a fraction of the network receiving the selfish miner blocks before a simultaneously block sent from another miner.
Sapirshtein et al. further extended the model to consider all possible actions a selfish miner can perceive. Block propagation time remains unassessed, since it is again considered to be much smaller than block generation time. Sapirshtein et al. also use the same notion of γ like Eyal and Sirer. Sapirshtein et al. model the whole process as a markov decision process. This markovian model was widely used and adopted in other research directions studying other aspects of selfish mining. Bai et al. extended the model even further to analyze multiple selfish miners. This resulted in a more complex state space of the markov decision process.
Xiao et al. study the impact on the profitability threshhold and revenue gain of a networking

advantage possessed by the selfish miner. They model the network as a graph and find that networking advantage correlates to the betweenness centrality of the selfish miner. Additionally it highly affects the profitability threshhold and revenue gain of the attacker. This indicates that the structure of the network influences the selfish mining strategy. However, this model remains very abstract, since only the peer-to-peer layer and structure is modelled as a graph, disregarding any limitations imposed by physical infrastructure such as bandwidth. Nonetheless, it indicates that the underlying network influences the blockchain overlay, strengthening the assumption that there is a highly influencial dependency between networking effects and selfish mining.

It is not contested by any of the previous research, that network capabilities and communication delay impact selfish mining [10], although most research model block propagation as instantaneous. Another factor is, that most research which is concerned with selfish mining, builts on top of the model presented by Sapirshtein et al. Both factors contribute to the negligence of networking effects, when analyzing selfish mining. Assuming that the underlying network does influence the system built on top, this master thesis aims to analyze the impact of networking effects on selfish mining. It is therefore important to represent the network in the model, which is used to analyze selfish mining.

2.2 Blockchain Network Models

Bitcoin and Proof-of-Work blockchains in general have been additionally modelled and analyzed from a networking perspective. In order to study selfish mining with the context of networking effects it is necessary to analyze the network. Most blockchain network models are concerned with the estimation of consistency. Consistency is the property of a blockchain that all honest parties output the same block sequence. Garay et al. study the core of the bitcoin protocol formally [5]. They analyze the protocol in a synchronous communication network and show persistence and liveness of committed transactions. Garay et al. further proof that the adversarial computational power bound to reach Byzantine Agreement is $1/2$ of the network for a synchronized network. The adversarial bound decreases as the network drifts further away from synchronization [5]. The Analysis of Garay et al. indicate that the network highly influences the behavior of Proof-of-Work blockchains.

Pass et al. propose a new network model to analyze blockchains in terms of consistency and liveness in an asynchronous network [11]. They do not make any assumptions of synchronicity and proof consistency in a network with with adversarial delays that are a-priori bounded. They show that the proof of work hardness needs to be set as a function of the maximum network delay. New peers joining the network or peers getting corrupted are also modelled. They proove that Nakamotos protocol satisfies consistency even in a network with message delays.

Kiffer et al. built on top of the models of Garay et al. and Pass et al., but formulate a simple markov chain based method to analyze consistency. Additionally they provide lower bounds for consistency. They also analyze the GHOST protocol, where consensus is built over the heaviest observed subtree, in addition to the longest chain rule [9]. The model is based on rounds of communication. The modelled adversary controls a fraction of honest peers and can delay and

reorder messages within a threshhold δ. The model therefore captures network attacks from an adversary, but disregards other networking effects.

Gervais et al. introduce a novel framework to analyse security and performance of blockchain [6]. They model how network and consensus parameters influence stale block rate, block propagation times, throughput and security. Stale blocks are blocks which do not end up in the longest chain. Selfish mining is modelled as a markov decision process. The network layer is characterized by block size and the information propagation mechanism. Gervais et al. simulate the system over a network consisting of point-to-point connections between peers. Those channels are defined by latency and bandwidth. Latency is set using global IP latency statistics. One major result is that an increasing block size increases block propagation time linearly and stale block rate exponentially.

Gopalan et al. utilize rumor-spreading to implement a new stochastic network model for blockchain [7]. They study stability and scalability of their model. Each peer communicates at a given rate his oldest blocks to his neighbors. Communication channels are also bandwidth limited. This setup introduces network delays to blocks, which depends on the instantaneous network congestion. Unlike previous stochastic network models Gopalan et al. do not introduce delay based on sampling data, but rather on the communication behavior of peers. Since network congestion depends on the behavior of peers and selfish mining is a deviating behavior, the model introduced by Gopalan et al. will be used in the following to analyze selfish mining and networking effects.´

Chapter 3

Model

In order to model networking effects and selfish mining, it is essential to capture network properties in an analytic model. The model can then be used to estimate selfish mining profitability. Gopalan et al. have introduced a new blockchain model, which captures network properties.

3.0.1 Bitcoin Mining Fundamentals

To understand selfish mining and its implications on network behavior it is essential to understand bitcoin mining in general. Bitcoin utilizes proof-of-work blockchain as a distributed ledger technology. It includes transactions into so called blocks. Blocks possess a unique ID and reference a previous block [13]. This construct builds a directed acyclic graph. The root of this tree is also called genesis block. Thus, every block directly or indirectly references the genesis block.
A correct block includes a nonce, which solves a cryptographic puzzle. The challenge is to alter the nonce until the hash of the set of transactions, the hash of the previous block and the nonce produce a partial hash collision. Essentially, the hash has to be smaller than some threshold value, which is also referred to as difficulty [13]. Thus, Bitcoin binds block creation to the computational ressources a peer possesses, since the partial hash collision can only be solved through trial and error. The correctness of the block is easily verifiable through third parties. Thus, Bitcoin ensures a fair leader election through this process.
Bitcoin uses a peer-to-peer network to propagate the mined blocks in the system. The network is unstructured as every peer tries to maintain a minimum of eight connections and performs neighbor discovery over DNS, IRC and asking neighbors [13]. Blocks are propagated over the peer-to-peer layer through flooding.
Once a miner mines a block through solving a cryptographic puzzle, he can publish the block and receives rewards through transaction fees and mining rewards. This provides an incentive to the miner to generate as many correct blocks as possible [2].
Consensus is established over the longest chain rule [2]. This means that the block ending the longest chain determines the state of the blockchain. This also implies that a miner only receives rewards, if his mined blocks are included in the main chain. Thus, a miner wants to produce as many correct blocks, that are part of the main chain, as possible. A protocol

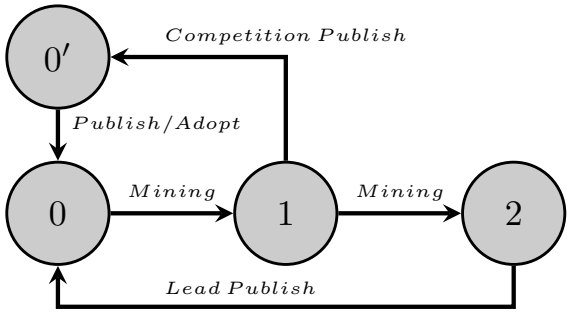

Figure 3.1: Abstract representation of state transtitions of eyal and sirer model for one selfish miner

maximizing reward gain is thus incentive compatible. A miner produces a relative share of blocks proportionally to his relative share of computational power of the whole network. Thus, a miner should produce a relative share of the main chain proportional to his relative share of computational power.
The original protocol, also called honest mining, assumes publishing blocks immediately after mining. Honest mining is assumed to be incentive compatible. It follows that no miner can earn disproportionate rewards by deviating from the protocol. Consequently, earning disproportionate rewards through deviation from the honest mining protocol, would disprove Bitcoin's incentive compatability claim.

3.0.2 eyal model

One protocol deviation is selfish mining, which was first introduced by [4]. Selfish Mining is a vulnerability, which aims at increasing revenue through block withholding. The selfish miner aims at producing a greater relative share of blocks of the main chain, than the relative share of computational power of the network. Therefore, selfish mining violates Bitcoins incentive compatibility claim, as it offers a more profitable mining protocol than honest mining. This is problematic, since it not only breaks fair leader election, but also results in potentially longer confirmation times for transactions of users. Eyal and Sirer model the network over a set of miners. A miner finds a subsequent block after a time interval that is exponentially distributed[4]. They further define the revenue of a miner as his fraction of total blocks on the longest chain. The selfish miner withholds mined blocks [4]. This selfish miner now possesses a private chain, which differs from the publicly known chain. Based on the difference between those two chains, the selfish miner performs actions. For clarification the state space and actions are modelled in 3.1. The numbers in the states indicate the lead of the private to the public chain. s denotes the lead of the private chain compared to the public chain. We can identify a total of four different actions.

- $Mining$: This action means that the peer has mined block. Mining adds the block to the private chain. It therefore causes s to increase.

- *Lead Publish*: When s increases to 2, the selfish miner will publish his private chain. It therefore causes s to change from 2 to 0.
- *Competition Publish*: When s is 1 and the selfish miner receives a block from another peer, he will publish his block of the same height from the private chain instead of the received one, to compete against the other miner. This causes a state transition to $0'$.
- *Publish*: If the selfish miner is in state $0'$, he is in a competition situation. The selfish miner will immediately publish his next mined block. This will cause the selfish miner to transition to state 0.
- *Adopt*: The selfish miner will adopt the main chain once he receives a new block in a competition situation.

3.0.3 Gopalan Model

The model of Gopalan et al. consists of a set of peers P connected through a peer-to-peer network. Peers add blocks to the blockchain through a process called mining.
The peer-to-peer network is modelled as an undirected Graph $H = (V, E)$. An edge $(i, j) \in E$ represents communication possibilities between $v_i \in V$ and $v_j \in V$. The set of vertices is finite, such that $|V| = N \in \mathbb{N}$. Vertices are associated with peers, such that v_i represents peer $p_i \in P$. Additionally, a directed acyclic graph $G_{p_i}(t) = (B_{G_{p_i}}(t), E_{G_{p_i}}(t))$ is associated with each peer p_i, at each point in time $t \in \mathbb{R}+$. The vertex set $B_{G_{p_i}}(t) \subset \mathbb{N}$ represents the blocks known of peer p_i at time t. The associated edge set of $E_{G_{p_i}}(t)$ represents references between blocks. The following holds true for shorter notations:

$$B_G(t) = \cup_{i=1}^{N} B_{G_{p_i}}(t) \text{ and } E_G(t) = \cup_{i=1}^{N} E_{G_{p_i}}(t) \tag{3.1}$$

Furthermore, the following equations hold for the principle of blockchains:

$$\forall p \in P : G_{p_i}(0) = (\{0\}, \emptyset) \tag{3.2}$$

$$t_1 < t_2 \rightarrow B_{G_{p_i}}(t_1) \subseteq B_{G_{p_i}}(t_2) \tag{3.3}$$

$$t_1 < t_2 \rightarrow E_{G_{p_i}}(t_1) \subseteq E_{G_{p_i}}(t_2) \tag{3.4}$$

Note that in this representation 0 denotes the genesis block described in equation 3.2.
$G_{p_i}(t)$ evolves over time. Blocks arrive over continuous time according to a stationary point process A with intensity λ. Each block $b \in \mathbb{N}$ arrives at a random peer p_i. This models peer p_i mining block b at time t and that at this time the block is also added to $B_{G_{p_i}}(t)$.
References are added to $E_{G_{p_i}}(t)$ according to policy and depending on $G_{p_i}(t^-)$, where t^- is a moment in time infinitesimally before t. O_i denotes the set of outgoing neighbors of block i.
The communication is modelled as a marked point process T_{p_i}. Each mark corresponds to another peer $p_j \in P \backslash \{p_i\}$. In an epoch peer p_i contacts p_j and thus, adds the lowest numbered block of $B_{p_i}(t) \backslash B_{p_j}(t)$ to the set of Vertices B_{p_j}. If $B_{p_i}(t) \backslash B_{p_j}(t)$ is not empty, E_{p_j} is also updated accordingly.

The peer-to-peer network dynamics are modelled as a continuous time rumor-spreading process with exogenous arrivals [7]. Since communication is bound to the process T_{p_i}, the block dissemination is bandwidth limited. Reference selection and thus O_{p_i} is chosen accordant to longest chain policies [7].
Let $L_{p_i}(t)$ denote the set of nodes farthest away from the genesis block 0, known to peer p_i at time t.

$$L_{p_i}(t) := \{j \in B_{p_i}(t) : d(j,0) \geq d(j',0), \forall j' \in B_{p_i}(t)\} \tag{3.5}$$

Note that the set $O_{p_i} \cap L_{p_i}(t)$ is non empty. This constructs a simple directed acyclic graph. The Tree Policy [7] can then be determined as $|O_{p_i}| = 1$ and establishes the following relationship:

$$|E_{G_{p_i}}(t)| = |B_{G_{p_i}}(t)| - 1 \tag{3.6}$$

Every block will have exactly one outgoing reference, according to some deterministic rule [7]. Gopalan et al. assume that the block with the lower index number will be chosen.

3.0.4 extension – selfish mining inclusion

The selfish mining attack is described as a peer executing a protocol deviant from honest mining [4]. Therefore a selfish miner can be modelled according to the model described in 3.0.3 through altering the reference selection and communication process. The reference selection process is policy driven, and can thus be modified by providing a new selfish policy.
Peer $SM \in P$ has an associated policy slightly different to 3.5. Note that to follow the Tree Policy [7], a deterministic rule has to be established for the case that $|O_{SM} \cap L_{SM}(t)| > 1$.
Assume that SM has the knowledge of the set of blocks mined through him, $M_{SM}(t) \subset B_{G_{SM}}(t)$. SM will set

$$(L_{SM}(t) \cap M_{SM}(t)) \neq \emptyset \rightarrow L'_{SM}(t) \subset (L_{SM}(t) \cap M_{SM}(t)) \tag{3.7}$$

It then follows that $|L'_{SM}(t)| = 1$. This modified tree policy sets references according to the original selfish mining protocol described by Eyal and Sirer.
The second aspect to be modified is the communication process. Key idea of selfish mining is block withholding. The selfish miner possesses three blockchain representations.

- $G_{SM_{public}}(t)$: which is updated by other peers.
- $G_{SM_{comm}}(t)$: which is used to update other peers.
- $G_{SM_{private}}(t)$: with the following relations:
  - $G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$.
  - $G_{SM_{private}}(t) \backslash G_{SM_{public}}(t)$ represents blocks mined but unpublished by the selfish miner.

The concept has been visualized in 3.2. A total number of five processes is used to let all entities interact with each other.

- *Arrival Process* A: Blocks arrive to the selfish miner over the external arrival process A.

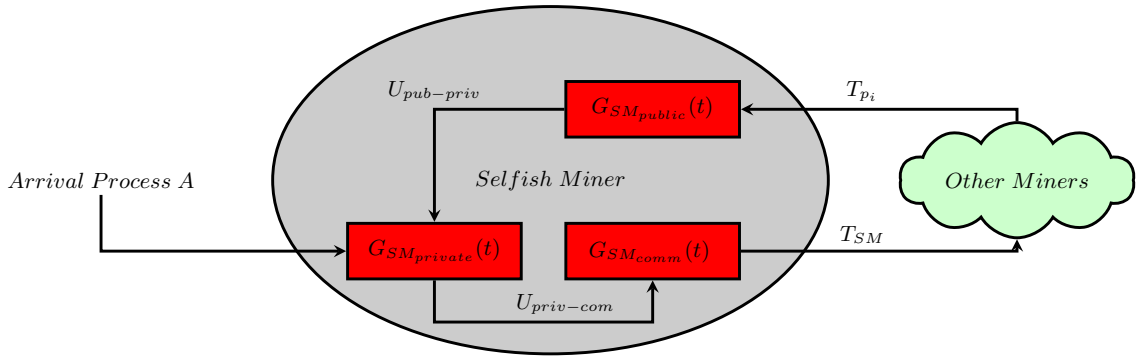

Figure 3.2: Abstract representation of model entities and communication processes

- T_{p_i}: Ensures blocks from other peers are communicated to $G_{SM_{public}}(t)$.
- $U_{pub-priv}$: Ensures that $G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$ holds true, meaning $U_{pub-priv}$ updates $G_{SM_{private}}(t)$, when new blocks arrive to $G_{SM_{public}}(t)$ from other peers.
- $U_{priv-com}$: Updates $G_{SM_{comm}}(t)$ according to $G_{SM_{private}}(t)$ and the selfish mining rules S.
- T_{SM}: Ensures other peers are updated with blocks from $G_{SM_{comm}}(t)$.

S is a set of rules which describes how $G_{SM_{private}}(t)$ updates $G_{SM_{comm}}(t)$. The rules have to follow the state description of Eyal and Sirer3.0.2. Therefore we need a state variable describing the difference between private and public chain. Let s be the state variable determining selfish mining actions [4]. Then s can be described as a difference between $G_{SM_{private}}(t)$ and $G_{SM_{public}}(t)$.

$$max\_dist(G_{p_i}(t)) := d(j,0), j \in L_{p_i}(t) \tag{3.8}$$

$$max\_dist\_mined(G_{p_i}(t)) := d(j,0), j \in M_{p_i}(t) \tag{3.9}$$

$$s(t) := max\_dist(G_{SM_{private}}(t)) - max\_dist(G_{SM_{public}}(t)) \tag{3.10}$$

$$s_{mined}(t) := max\_dist\_mined(G_{SM_{private}}(t)) - max\_dist(G_{SM_{public}}(t)) \tag{3.11}$$

Let t_{inc} refer to the set of times, where s increased and analogous t_{dec} refer to the set of times, where s decreased. Additionally, let t'_{inc} refer to the set of times, where s_{mined} increased and analogous t'_{dec} refer to the set of times, where s_{mined} decreased. Note that $s > 0$, because $G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$. Let $f_{-1}(t)$ be a function that outputs the point in time, where s changed the latest before t. $U_{priv-com}$ can then be characterized through four kind of update actions. Analogous to Subsection3.0.2, those actions are $Lead\ Publish$, $Competition\ Publish$, $Publish$ and $Adopt$. $Mining$, the fifth action described in Subsection3.0.2, is modelled through the arrival process. This can be used to model the selfish mining protocol desribed by Eyal and Sirer.

1. $Lead\ Publish$: Assume $t \in t_{inc}$ and $s(t) \geq 2$, then $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that $G_{SM_{comm}}(t) = G_{SM_{private}}(t)$.

2. *Competition Publish*: Assume $t \in t_{dec}$, $s(t) = 0$, $s(f_{-1}(t)) = 1$, $s(f_{-1}(t)^-) = 0$. This means that the selfish miner mined a block, did not publish it and now received a block from another of the same height. This leads to the competition scenario. Accordingly, $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that it includes the subgraph induced by the nodes on the paths between $L'_{SM}(t)$ and 0. This transitions to

$$0'(t) \rightarrow \big(t \in t_{dec} \wedge s(t) = 0 \wedge s(f_{-1}(t)) = 1 \wedge s(f_{-1}(t)^-) = 0\big) \tag{3.12}$$

3. *Publish*: Assume $0'(t^-) = \top$ and $t \in t_{inc}$, $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that it includes the subgraph induced by the nodes on the paths between $L'_{SM}(t)$ and 0.

4. *Adopt*: Assume $0'(t^-) = \top$ and $s'(t) = -1$, then $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that $G_{SM_{comm}}(t) = G_{SM_{private}}(t)$.

Chapter 4

Contribution

T Therefore, the model proposed by Gopalan et al. has been enhanced to model selfish mining behaviour in 3.0.4. The relationship between selfish mining and networking effects can be charactized by a number of key questions. Those questions can be split up in two groups. The first group considers how the network influences selfish mining. Key aspects include:

1. Xiao et al. [14] show in their model that revenue gain and profitability threshhold correlates to betweenness centrality. Does this correlation also show 3.0.4?
2. Does a networking advantage increase revenue gain and profitability threshhold?
3. Does a certain network topology influence selfish mining effectiveness?

The second group considers how selfish mining influences the network. Key aspects include:

1. Does the network have a different throughput, if one peer is executing the selfish mining protocol?
2. Does the network have a different block propagation time, if one peer is executing the selfish mining protocol?
3. Does the network show different congestion peeks, if one peer is executing the selfish mining protocol?
4. If the network shows congestion peeks, does it correlate to certain actions the selfish miner is performing?

´

Chapter 5

Evaluation

Chapter 6

Conclusion

Bibliography

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong. A deep dive into blockchain selfish mining. Cryptology ePrint Archive, Report 2018/1084, 2018. https://eprint.iacr.org/2018/1084.

[2] S. Barber, X. Boyen, E. Shi, and E. Uzun. Bitter to better — how to make bitcoin a better currency. In A. D. Keromytis, editor, *Financial Cryptography and Data Security*, pages 399–414, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-32946-3.

[3] J. R. Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[4] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. *CoRR*, abs/1311.0243, 2013. URL http://arxiv.org/abs/1311.0243.

[5] J. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[6] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16, 2016.

[7] A. Gopalan, A. Sankararaman, A. Walid, and S. Vishwanath. Stability and scalability of blockchain systems, 2020.

[8] O. Ibe. *Markov processes for stochastic modeling*. Newnes, 2013.

[9] L. Kiffer, R. Rajaraman, and A. Shelat. A better method to analyze blockchain consistency. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 729–744, 2018.

[10] T. Leelavimolsilp, L. Tran-Thanh, and S. Stein. On the preliminary investigation of selfish mining strategy with multiple selfish miners. *CoRR*, abs/1802.02218, 2018. URL http://arxiv.org/abs/1802.02218.

[11] R. Pass, L. Seeman, and A. Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 643–673. Springer, 2017.

[12] A. Sapirshtein, Y. Sompolinsky, and A. Zohar. Optimal selfish mining strategies in bitcoin. *CoRR*, abs/1507.06183, 2015. URL `http://arxiv.org/abs/1507.06183`.

[13] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys and Tutorials*, 18:1–1, 03 2016. doi: 10.1109/COMST.2016.2535718.

[14] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou. Modeling the impact of network connectivity on consensus security of proof-of-work blockchain, 2020.