



# 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.
It 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 [7]. 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 [7]. 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 [7]. 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 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.
One protocol deviation is selfish mining, which was first introduced by [3]. 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.
Studying the impact of selfish mining and other mining protocol deviations is necessary, because without proper risk assessment no effective countermeasures can be implemented.

# 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, to estimate realistic results.
Blockchain Mining is most commonly modelled through markov decision processes. A markovian process is a discrete time stochastic control process. It is generally used to model decision making, where the outcomes are influenced by random processes and the decision of the decision maker. 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. The authors model the network over a set of miners. A miner finds a subsequent block after a time interval that is exponentially distributed. Eyal and Sirer further define the revenue of a miner as his fraction of total blocks on the longest chain. The selfish miner withholds mined blocks [3]. 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 2.1. The numbers in the states indicate the lead of the private to the public chain. M denotes mining a block, P publishing the block because of leading by two blocks. CP is an action triggered in state 1 when the miner receives a block. This block will have the same height as the block previously mined by the selfish miner. The selfish miner will compete against this received block by publishing his own. This will cause the selfish miner to transition to 0'. C denotes the transition back to 0. The selfish miner will publish his next mined block immediately or will continue to mine on the next received block in C causing him to transition from 0' to 0.
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 [3]. 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 $\gamma$ [3]. $\gamma$ resembles a fraction of the network receiving

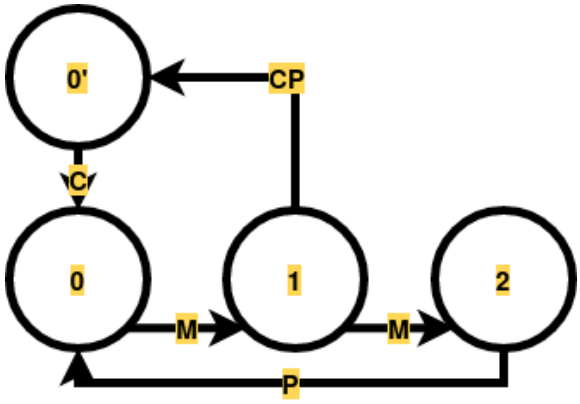


Figure 2.1: State Space representation of selfish mining

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 $\gamma$ 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.
It is not contested by any of the previous research, that network capabilities and communication delay impact selfish mining [5], 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.
Xiao et al. [8] 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.

# 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.

**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[1]:

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

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 $\lambda$. 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 $i$.

[1] aber warum eigentlich? macht die zeitabhängigkeit das ganze nicht kaputt?

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 [4]. 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 [4].

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 [4] 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 [4]. Gopalan et al. assume that the block with the lower index number will be chosen.

**model extension – selfish mining inclusion** The selfish mining attack is described as a peer executing a protocol deviant from honest mining [3]. Therefore a selfish miner can be modelled according to the model described in 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 [4], 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. As such the selfish miner not only possesses a blockchain representation of the form $G_{p_i}(t) = (B_{G_{p_i}}(t), E_{G_{p_i}}(t))$, but rather $G_{p_i}(t) = G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$, where $G_{SM_{private}}(t) \backslash G_{SM_{public}}(t)$ represents blocks mined but unpublished by the selfish miner.

The concept has been visualized in 3.1.

$G_{SM_{comm}}(t)$, a third representation, is used to update other peers just as described in 3. $G_{SM_{private}}(t)$ interacts with $G_{SM_{public}}(t)$ through an instantaneous update process $U$. The first task of $U$ is to ensure that $G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$ holds true, meaning $U$ updates $G_{SM_{private}}(t)$, when new blocks arrive to $G_{SM_{public}}(t)$.

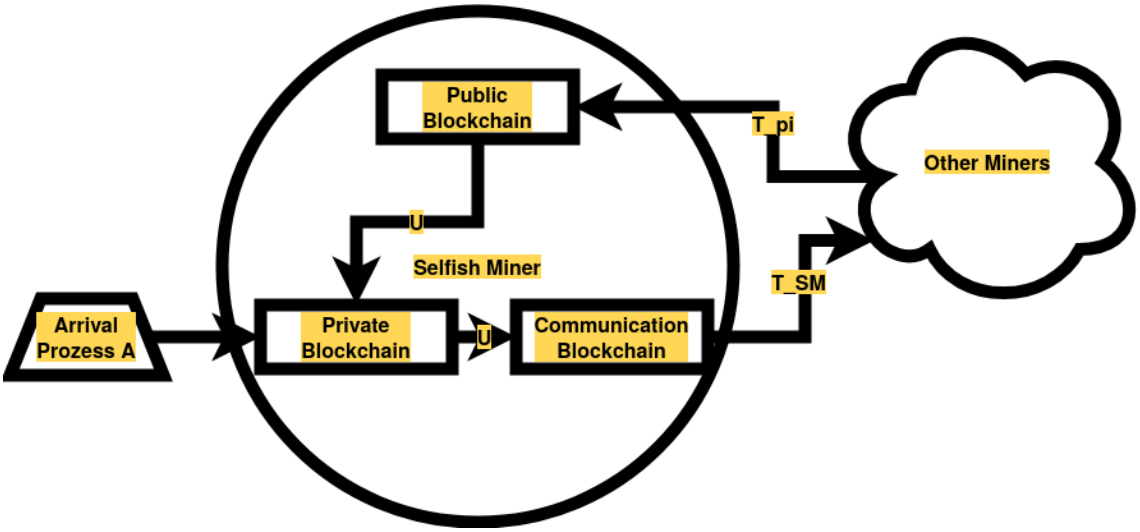


Figure 3.1: Abstract representation of model entities and communication processes

The second task of $U$ is updating $G_{SM_{comm}}(t)$ according to $G_{SM_{private}}(t)$, which is the heart of the selfish mining protocol described by [3].
Let $s$ be the state variable determining selfish mining actions [3].
Then $s$ can be described as a difference between $G_{SM_{private}}(t)$ and $G_{SM_{public}}(t)$.

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

$$s(t) := max\_dist(G_{SM_{private}}(t)) - max\_dist(G_{SM_{public}}(t)) \tag{3.9}$$

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. Let $f_{-1}(t)$ be a function that outputs the point in time, where s changed the latest before $t$. $U$ can then be characterized through three kind of update actions. This can be used to model the selfish mining protocol desribed by Eyal and Sirer.

1. Assume $t \in t_{inc}$ and $s(t) \geq 2$, then $U$ updates $G_{SM_{comm}}(t)$, such that $G_{SM_{comm}}(t) = G_{SM_{private}}(t)$.
2. Assume $t \in t_{dec}$ and $s(t) = 0$, then $U$ updates $G_{SM_{comm}}(t)$, such that it includes the subgraph induced by the nodes on the paths between $L'_{SM}(t)$ and 0.
3. Assume $t \in t_{dec}$ and $s(t) = -1$, then $U$ updates $G_{SM_{comm}}(t)$, such that $G_{SM_{public}}(t) \subseteq G_{SM_{comm}}(t)$.
4. Assume $t \in t_{inc}$, $s(t) = 1$, $s(f_{-1}(t)) = 0$, $s(f_{-1}(t)^-) = 1$, then $U$ updates $G_{SM_{comm}}(t)$, such that it includes the subgraph induced by the nodes on the paths between $L'_{SM}(t)$ and 0.

[2] $T_{SM}$ represents the outgoing communications of the selfish miner. Since those are highly influenced by the receiving ends the selfish miner can not increase his communication capabilities bound to $T_{SM}$ in order to gain an advantage. However, influencial changes based on the structure of $H$ and the position of the selfish miner will be analyzed in the latter.

[2] kann man, darf man $T_{SM}$ modifizieren? Ich würde argumentieren nein

# Chapter 4

# Contribution

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. 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. Therefore, the model proposed by Gopalan et al. has been enhanced to model selfish mining behaviour in 3. 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. [8] show in their model that revenue gain and profitability threshhold correlates to betweenness centrality. Does this correlation also show 3?
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] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.