FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Merged Mining: Analysis of Effects and Implications

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering and Internet Computing

eingereicht von

Alexei Zamyatin, BSc
Matrikelnummer 1168338

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Mitwirkung: Dipl.-Ing. Aljosha Judmayer

Wien, 24. August 2017

Alexei Zamyatin

Edgar Weippl

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Merged Mining: Analysis of Effects and Implications

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Alexei Zamyatin, BSc
Registration Number 1168338

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Assistance: Dipl.-Ing. Aljosha Judmayer

Vienna, 24th August, 2017

Alexei Zamyatin

Edgar Weippl

Erklärung zur Verfassung der Arbeit

Alexei Zamyatin, BSc
Franz-Sillergasse 6, A-1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 24. August 2017

Alexei Zamyatin

Ergebnisse dieser Diplomarbeit, getitelt „Merged Mining: Analysis of Effects and Implications“, wurden in folgender Publikation veröffentlicht:

A. Judmayer, A. Zamyatin, N. Stifter, A. Voyiatzis, E. Weippl. „Merged Mining: Curse or Cure?“ 1st International Workshop on Cryptocurrencies and Blockchain Technology, 2017

Hiermit wird von den Mitautoren bestätigt, dass Alexei Zamyatin alle aus der oben genannten Publikation in diese Diplomarbeit übernommenen Textpassagen selbstständig verfasst hat.

Wien, 24. August 2017

Aljosha Judmayer
Nicholas Stifter
Artemios Voyiatzis
Edgar Weippl

Danksagung

An dieser Stelle möchte ich mich besonders bei Aljosha Judmayer bedanken, der mich während meiner Diplomarbeit betreut und umfangreich unterstützt hat. Des Weiteren bedanke ich mich bei Nicholas Stifter, Artemios Voyiatzis und Edgar Weippl für deren großartige Hilfe bei der Veröffentlichung der Ergebnisse dieser Arbeit. Ebenfalls bedanke ich mich bei Georg Merzdovnik und Philipp Schindler für die vielen Anregungen und wertvollen Diskussionen. Besonderer Dank gilt außderdem meinen Eltern, die mir dieses Studium ermöglicht haben und immer mit Rat und Tat zur Seite gestanden sind, sowie meiner Freundin Erika, die mich im Verlauf des Studiums immer unterstützt und ermutigt hat.

Kurzfassung

Merged Mining beschreibt das Konzept von parallelem Mining auf mehreren verschiedenen Kryptowährungen, ohne zusätzlichen Einsatz von Rechenleistung für Proof-of-Work. Bei dessen Einführung 2011 war Merged Mining vor allem als Starthilfe für neue Kryptowährungen angedacht und sollte die Fragmentierung von Rechenleistung auf konkurrierende Systeme verhindern. Obwohl Merged Mining bereits in einer signifikanten Zahl an Kryptowährungen eingesetzt wird, ist bisher wenig über dessen Auswirkungen auf die darunterliegenden Systeme bekannt.

In dieser Diplomarbeit wird, unter Durchführung der ersten umfangreichen Analyse der praktischen Aspekte von Merged Mining, Aufschluss über dessen Auswirkungen auf Kryptowährungen gegeben. Im Zuge dieser Studie wird, zwecks zuverlässiger Evaluierung von Zentralisierungserscheinungen in Kryptowährungen, ein Schema zur Attributierung generierter Blöcke zu deren ursprünglichen Minern vorgestellt. Die Ergebnisse dieser Arbeit zeigen auf, dass Mining Pools signifikante Anteile der Rechenleistung in Kryptowährungen, welche Merged Mining einsetzen, akkumulieren und über lange Zeitraume kontrollieren konnten. In manchen Fällen wurden dadurch die Sicherheitsgarantien des am Kern dieser Systeme liegenden Nakamoto Konsensus ungeltend gemacht. Wir evaluieren die daraus ableitbaren Sicherheitsprobleme für Kryptowährungen und stellen diese Entwicklungen den geplanten Effekten von Merged Mining gegenüber.

Abstract

Merged mining refers to the concept of mining more than one cryptocurrency without necessitating additional proof-of-work effort. Merged mining was introduced in 2011 as a boostrapping mechanism for new cryptocurrencies and countermeasures against the fragmentation of mining power across competing systems. Although merged mining has already been adopted by a number of cryptocurrencies, to this date little is known about the effects and implications.

In this thesis, we shed light on this topic area by performing a comprehensive analysis of merged mining in practice. As part of this analysis, we present a block attribution scheme for mining pools to assist in the evaluation of mining centralization. Our findings disclose that mining pools in merge-mined cryptocurrencies have operated at the edge of, and even beyond, the security guarantees offered by the underlying Nakamoto consensus for extended periods. We discuss the implications and security considerations for these cryptocurrencies and the mining ecosystem as a whole, and link our findings to the intended effects of merged mining.

Contents

CHAPTER 1

Introduction

Bitcoin was introduced as the first decentralized ledger currency in 2008. Since then, the field of cryptocurrencies has experienced a rapid growth in popularity, both from the academic and private sectors. Today, over 800 blockchain-based digital currencies and assets are listed on numerous exchanges [1], while new proposals and implementations are introduced on a daily basis. However, despite its popularity, the blockchain technology still faces numerous unresolved problems in terms of performance, scalability, sustainability, security and decentralization.

Merged mining refers to the process of searching for proof-of-work (PoW) solutions for multiple cryptocurrencies concurrently without requiring additional computational resources. The rationale behind merged mining lies in leveraging on the computational power of different cryptocurrencies by bundling their resources instead of having them stand in direct competition, and also to serve as a bootstrapping mechanism for small and fledgling networks [59, 72]. Despite having been introduced shortly after Bitcoin and having been actively employed by a number of cryptocurrencies for several years, this concept has recently received little attention from the scientific community. Nevertheless, new and emerging cryptocurrencies such as *Rootstock* continue to consider and expand on the concept of merged mining in their designs to this day [44].

In the past, concerns have been voiced that merged mining could possibly lead to additional security risks and challenges [59]. In particular, the realistic threat of network centralization has rendered merged mining a controversial topic in the Bitcoin community. Ali et al. [3] observed a critical level of mining centralization in the merge-mined cryptocurrency *Namecoin*, concluding that merged mining is failing in practice. These alarming findings were not the result of direct investigations into merged mining itself, but rather emerged as part of a report on the experiences with the real-world deployment of a decentralized PKI service on top of the Namecoin blockchain. Hence, an in-depth analysis of merge-mined cryptocurrencies based on real-world data is necessary to determine if such observed failures in practical applications are systemic to the underlying concept of merged mining.

In this thesis we conduct the first extensive study on the impacts of merged mining on individual cryptocurrencies. We discuss security implications and considerations regarding merged mining, while relating previous arguments from [59] to the results of our study. We seek to provide empirical evidence either confirming or falsifying these arguments and extend the discussion by providing ideas and examples for future experiments, which can lead to a better understanding and classification of merged mining.

To cover a broad spectrum of merge-mined cryptocurrencies we analysed two established players and pioneers of the field, namely Namecoin and Dogecoin, as well as two relatively young merge-mined cryptocurrencies supporting merged mining with more than one PoW algorithm, namely Huntercoin [32] and Myriadcoin [54]. Thereby, we present the following contributions:

- We provide a technical explanation of the merged mining concept and model the process as conducted by involved miners.
- We analyze the effects and implications of merged mining in four cryptocurrencies over time and comment on its adoption, the related difficulty increase, as well as other characteristic patterns.
- We introduce a deterministic mapping scheme that attributes blocks to specific miners and mining pools.
- We provide empirical evidence for centralization risks in cryptocurrencies involved in merged mining. Furthermore, we are successful in attributing merged mining activity to an apparently small set of mining pools.
- We discuss the related security implications for cryptocurrencies implementing merged mining.
- Concluding, we outline possible improvements of the concepts of merged mining to help mitigate potential security threats.

The remainder of this thesis is structured as follows. In Chapter 2 we provide the necessary background information on fundamental concepts regarding proof-of-work based cryptocurrencies and deliver an overview of the state of the art in the context of proof-of-work reusing and the inner workings of merged mining. Chapter 3 describes the experimental methodology, as well as the cryptocurrencies considered in our analysis. In Chapter 4 we present the results of our study. Chapter 5 provides discussions the security implications in relation to theoretical arguments regarding merged mining and gives an outlook on potential improvements. Furthermore, a critical reflection of the performed work is conducted. Finally, we propose new research questions and conclude the thesis in Chapter 6, pointing out interesting directions for future work.

CHAPTER 2

State of the art

In this chapter we provide the necessary background information on Bitcoin and the general concept of proof-of-work blockchains (Section 2.1) and give an introduction to the concept of proof-of-work reusing, including an in-depth description of merged mining and an overview of other proof-of-work reusing mechanisms (Section 2.2).

2.1 Bitcoin and Proof-of-Work Blockchains

Bitcoin was introduced as the first decentralized peer-to-peer (P2P) cryptocurrency in 2008 by an author under the pseudonym Satoshi Nakamoto [55]. Since its introduction, Bitcoin has emerged as the most popular digital medium of monetary exchange throughout the past years and, at the moment of writing, maintains a market capitalization of over 50 billion USD [1].

A key aspect which distinguishes Bitcoin from earlier cryptocurrency proposals [9,17,25,75] is its novel distributed consensus approach which removes the necessity of a trusted third party for agreeing on a common transaction ledger. This consensus mechanism, generally termed *Nakamoto consensus*, leverages on proof-of-work (PoW) puzzles and a data structure called the *blockchain* to achieve eventual agreement on the set and ordering of transactions by an anonymous and changing set of participants, thereby facilitating decentralized or so-called *permissionless* cryptocurrencies. The blockchain is an append-only ledger of digitally signed transactions which are grouped in *blocks chained* together via the hashes of their predecessors. In Bitcoin, a transaction consists of a series of inputs from and outputs to user accounts. An account is thereby represented by a hash over the user's public key and there exists no limit to the number of accounts a user can maintain.

The PoW acts as a form of *key-less signature* to authenticate the new blocks and in turn the blockchain as a whole [62]. This *implicit consensus* process can be described as a "random leader election" on each solved PoW, where the leader is allowed to propose a new block and implicitly agrees on the preceding block by appending his new block to the end of the blockchain [59].

The term *blockchain* is also commonly used as a broad descriptor for modern cryptocurrencies inspired by Bitcoin, even if the underlying consensus mechanism does not rely on PoW or indeed the utilized data structure actually differs from the blockchain construction.

2.1.1 Proof-of-Work and Mining

The validity of blocks and the blockchain is determined by both agreed upon protocol rules as well as a valid proof-of-work over block data that conforms to a certain *difficulty*. Participants in the network will consider the longest consecutive chain of blocks with the most cumulative difficulty, starting from an agreed upon *genesis block*, as the valid blockchain. The process by which participating nodes in Bitcoin and similar proof-of-work cryptocurrencies search for valid PoW puzzle solutions and thereby partake in the consensus process is referred to as *mining*, and the speed at which such *miners* find solution candidates for the PoW is called *hash rate*.

Mining represents the process of solving and disseminating *cryptographic* or *scratch-off-puzzles* [52] as a means of PoW. The PoW puzzle used in Bitcoin is a partial pre-image attack on the SHA256 hash function [29], i.e., miners must find a combination of the current block header and a random nonce such that the resulting double SHA256 *block hash* lies below a predefined target T. This target is expressed in form of an alphanumerical 256-bit value with a specified number of leading zeros. The more zeros are required to be at the beginning of the target and hence the actual block hash, the smaller the set of existing PoW solutions. Hence, a small target T means that it is difficult to find a valid solution to the PoW puzzle, i.e., a miner is expected to perform more attempts. Due to the implementation-specifics of Bitcoin, the maximum possible value for the target is $T_{\max} = 2^{224}$, i.e., 32 leading zeros, which results in 2^{32} expected attempts to find a valid solution. As mentioned, the hardness of a cryptocurrencies PoW puzzle is often expressed in terms of the *difficulty* D, defined as the ratio of the maximum target $T_{\max}$ to the current target $T_{current}$ [35]:

$$D = \frac{T_{\max}}{T_{current}} \tag{2.1}$$

In Bitcoin, the process of mining can be described as follows: miners collect all transactions they receive over the peer-to-peer network and consequently attempt to solve the current PoW puzzle. More detailed, miners iteratively search the problem space defined by the block header and the random nonce used as input, for hash collisions fulfilling the required target T. Each time a miner succeeds in finding a such valid PoW solution, he creates a new block and publishes it in the network, propagating the block to all known nodes. This way, new block is appended to the blockchain[1], implicitly increasing the security and

[1] Or, more accurately, the version of the blockchain of each node, which accepts the block as valid.

immutability guarantees of all previous blocks. As reward for the invested computational effort, the miner is granted newly generated or *minted* units of the underlying currency.

Miners may choose to join or leave the network at any time, thereby increasing or decreasing the overall mining power. In order to susain a more or less constant interval between consecutive blocks, the difficulty of the PoW puzzle must be adjusted dynamically. In Bitcoin, where the block interval is set to be approximately 10 minutes, the PoW difficulty is updated every 2,016 blocks, i.e. approximately every two weeks. This is achieved by selecting a new target T_{new} based on the current target $T_{current}$ and the elapsed time t since the last block:

$$T_{new} = \frac{T_{current} \cdot t}{I \cdot t_{target}} \tag{2.2}$$

where I is the difficulty adjustment interval, i.e. 2,016 blocks in Bitcoin, and t_{target} is the targeted block interval, i.e. 10 minutes in Bitcoin.

2.1.2 Security of Proof-of-Work Blockchains

The concept of proof-of-work not only enforces agreement on a consistent state of the global ledger, as part of the Nakamoto consensus, but also provides certain security and immutability guarantees. Due to its high computational complexity and resource intensiveness, PoW acts as a defense mechanism against so called *Sybil attacks*, where a small number of entities counterfeit multiple identities in the attempt to compromise large parts of the system [21]. Furthermore, every block secured by PoW and appended to the tip of the blockchain makes it more difficult for adversaries to perform changes to previously included transactions, as described in the example below.

Assume Alice wants to revoke a transaction she made in the past, e.g. in block B_i. To do so, she will have to recalculate the hash $H(B'_i)$, i.e. re-solve the PoW puzzle of the now modified block B'_i [2]. However, since block B_{i+1} includes a reference to the hash of the original block $H(B_i)$, Alice is forced to re-compute $H(B'_{i+1})$ as well. She must continue this process, until she reaches the head of the blockchain B_n. However, since the rest of the network, here assumed to act honestly, continues to generate and append new blocks to the blockchain, Alice must perform all mentioned calculations *faster* than all honest miners. The probability of success hence depends on the ratio of Alice's computational resources to the rest of the network: if Alice is able to accumulate a majority (i.e. more than 50%) of the mining power, the attack is expected to be successful with probability 1 given enough time.

The security properties of PoW cryptocurrencies are generally derived from the assumption that the majority of the overall mining power belongs to honest miners [78]. Early work in Bitcoin security modeling concluded that the mining power of all honest miners has to

[2] Changing any part of any transactions in a block, will modify the reference to the transactions (i.e., the root of the Merkle tree storing hashes of all transactions as leaves [13, 14]) in the block header, which in turn will result in a different block hash

be strictly greater than 50% to sustain the security of the blockchain [55, 69]. Should adversaries accumulate the majority of mining power, they can control the insertion of new transactions, the transaction fee market, and the supply of newly-mined coins, as well as potentially revert already recorded transactions and perform *double spending* attacks [39, 63, 68].

Attack strategies which can be successful even without controlling the majority of mining power, most notably *selfish mining* [24, 71] and *eclipse* attacks [28, 31, 60] have been the topic of recent work.

When performing selfish mining, an adversary creates a private chain parallel to the public blockchain, thereby attempting to stay ahead of the rest of the network. Each time the attacker finds a new block, he appends it to his private chain, instead of broadcasting it to the public. Then, depending on the ratio of the length of his private chain to the public blockchain, the attacker decides as whether to publish his chain or to continue the race against the other miners. Assume the attacker manages to overtake the honest miners, i.e. his private chain is longer than the public blockchain. If the honest miners are unable to catch up, the attacker continues to build his private chain, thereby earning all block rewards and transaction fees. However, if the public blockchain is about to draw even with the length of the attacker's private chain, he will broadcast his blocks to the network. By protocol definition, the rest of the network will now discard their current view of the blockchain and accept the attackers longer private chain as the new global state. As the other nodes will now need time to validate and re-index the enforced new state of the blockchain, the attacker gains a headstart on mining the next block, increasing his chances to further stay ahead of the honest miners.

Eclipse attacks, on the other hand, aim at isolating selected nodes of the network, thereby partitioning it into disconnected clusters. Assume Alice wants to perform a such attack on Bob. She then will attempt to control all or the majority of the nodes Bob is connected to in the network. If successful, Alice will be able to enforce any selected state of the blockchain on Bob, as he will have no possibility of verifying its correctness and consistency with the public blockchain. Furthermore, Alice would be able to filter Bob's transactions before forwarding them to other nodes.

The success probability of such adversarial strategies depends on the mining power share (α), as well as the network connectivity (γ) of the adversary [24, 60]. While, for example, a poorly connected attacker ($\gamma \approx 0.1$) is shown to require $\alpha > 0.33$ to successfully perform selfish mining attacks, an adversary connected to half of the nodes in the network ($\gamma \approx 0.5$) only requires $\alpha > 0.25$. Hence, in a conservative analysis, successful attacks on PoW cryptocurrencies are more likely when dishonest entities control more than 25% of the total mining power. An attacker able to accumulate a majority of the mining power ($\alpha > 0.5$) or control the majority of the nodes in the network ($\gamma > 0.5$), will be able to attempt adversarial strategies with very high probability of success.

2.1.3 Mining Pools

In the past years, mining in Bitcoin and other cryptocurrencies has gained on popularity, as it became more profitable. As the number of participating miners and hence the overall present hash rate increases, so does the PoW difficulty. The introduction of hardware dedicated to high performance mining operations (ASICs) even further accelerated this development [76]. This in turn has negative impacts on the variance of payouts of miners, leading to a trend towards collusive mining activities, as will be discussed in the following paragraphs.

The interval between each two consecutive blocks is exponentially distributed with mean D/h_i, where h_i denotes the hash rate of a miner i. Hence, the number of blocks found per time period by a single miner i follows a Poisson distribution with rate parameter

$$\lambda = \frac{h_i}{D} \tag{2.3}$$

From this, the miner's expected revenue can be derived as

$$E[R_i] = \frac{R_b h_i}{D} \tag{2.4}$$

where R_b represents the expected mining reward[3] and transaction fees of a block. The variance of revenue is then defined by

$$Var[R_i] = \frac{R_b^2 h_i}{D}. \tag{2.5}$$

As the variance of payouts is directly related to the invested hash rate individually acting or *solo* miners usually face highly irregular payout intervals [4]. Hence, aiming to achieve a more stable stream of revenue, miners collude to form so called *mining pools* [45, 73]. Thereby miners bundle their computational resources and share the received rewards in accordance to their contribution and based on some set of predefined rules of the pool. A mining pool can hence be described as a "*pool manager and a cohort of miners*" [23].

The mining pool operator, apart from maintaining the pool's servers, is responsible for fairly distributing the earned revenue among participating miners. Hence, the operator must estimate each miner's contribution to the mining operations of the pool. This is usually achieved by requiring miners to solve PoW puzzles at a lower difficulty, than required by the network. The solutions to these easier puzzles are referred to as *shares* and are submitted by all miners to the mining pool. Each share can thereby also represent a solution to the cryptocurrencies' current PoW puzzle, if it meets the required target. The submitted shares are used to estimate each miner's performed work and hence his

[3] Currently 12.5 BTC in Bitcoin.

[4] Assuming they have bounded resources and hence are unable to accumulate significant shares of the overall mining power.

portion of the block rewards and transaction fees earned by the pool. To compensate the administrative effort, the mining pool keeps a small proportion of the total revenue as a fee[5].

We note that the pool's PoW puzzle must be significantly easier than the actual PoW problem. Otherwise, the pool operator only will be able to estimate the work of the one miner who found a solution meeting the network's PoW target and hence created the next block. As a result, only this lucky miner would receive all rewards, which in turn would resemble solo mining. In practice, the operator will set the PoW target required by shares dynamically, based on each miner's hash rate: if large miners would receive too easy PoW problems, they would submit shares at high frequency increasing the load to the pool's servers.

Different reward distribution policies and related game-theoretic aspects are studied in [45, 68, 73], while adversarial strategies related to mining pools are discussed in [23, 30, 60, 71]. Furthermore, pool managers can maliciously mislead their miners into participating in attacks, as seen in the case of the Eligius pool launching an attack on Coiledcoin[6]. While taking such actions may result in miners switching to another pool once they learn about the attack, the delay of such consequences however may be enough for the pool to complete the attack. Therefore, one can consider the hash rate of a pool being controlled by a single entity.

2.1.4 Sustainability and Future Outlook of Proof-of-Work

Proof-of-work is a well established concept in terms of providing security and immutability in blockchains. However, its implementation is not without controversy. Although the resource heavy computations required by PoW increase the cost of attacks, they hinder the performance and scalability of the underlying system [10, 11, 78] and result in a significant power consumption [61]. Furthermore, while initially any participant could successfully participate in the mining process, rising competition resulted in the creation of pooled mining which in turn leads to centralization of computational resources [27]. The introduction of ASICs, i.e., hardware specifically constructed to perform mining operations [76], further amplified this development. As a result, the majority of computational power in Bitcoin and other cryptocurrencies is concentrated in a small set of mining pools.

While efforts towards replacing the resource-intensive mining process have so far yielded various promising approaches such as [12, 41, 50], their viability in practice is yet to be tested at a larger scale. Furthermore, due to the high degree of adoption of proof-of-work in various cryptocurrencies and the difficulties related to changing this consensus critical component, it can be assumed that PoW will remain an integral part of the overall cryptocurrency landscape in the foreseeable future.

[5] Usually between 1 and 5%.

[6] cf. `https://bitcointalk.org/index.php?topic=56675.msg678006#msg678006`

2.2 Reusing Proof-of-Work

The general idea of reusing proof-of-work such that the computational effort invested may also serve to verify a separate computation was first introduced by Jakobsson and Juels under the term *bread pudding protocols* in 1999 [34] and patented in 2008 [33]. The selected terminology points towards the main idea of the scheme: reuse computation by-products to minimize wasted resources. In the context of proof-of-work, this means to recycle unused or *stale* computations and utilize them as proof-of-work for other tasks.

It is, however, necessary to differentiate between the concept of *reusing* the properties of the performed proof-of-work in a self-contained manner and without modification thereof, and approaches which introduce new proof-of-work algorithms capable of fulfilling tasks other than securing consensus in a permissionless system. We refer to the latter concept as *proof-of-work re-purposing* and provide an overview of such mechanisms in Section 2.3.

In the following sections we describe how the idea of proof-of-work reusing is applied to proof-of-work cryptocurrencies by the concept of *merged mining* and provide an overview of other relevant proof-of-work reusing mechanisms.

2.2.1 Merged-Mining

Merged mining refers to the process of reusing (partial) PoW solutions from a *parent* cryptocurrency as valid proofs-of-work for one or more *child* cryptocurrencies. It was introduced as a solution to the fragmentation of mining power among competing cryptocurrencies and as a bootstrapping mechanism for small networks. Merged mining was first implemented in Namecoin in 2011, with Bitcoin acting as the parent cryptocurrency. One of the earliest descriptions of the mechanism as it is used today was presented by Satoshi Nakamoto in [72]. Apart from the source code of the respective cryptocurrencies implementing merged mining, a technical explanation is given in the Bitcoin Wiki [56].

For a parent cryptocurrency to allow merged mining it must fulfill only one requirement: it must be possible to include arbitrary [7] data within the input over which the proof-of-work in the parent is established. The main protocol logic of merged mining in turn resides in:

- The specification and preparation of the data linked to (or included in) the block header of the parent, e.g., a hash of the child block header.
- The implementation of the verification logic in the client of the child blockchain, i.e., the child blockchain must be able to verify the PoW of the merge-mined blocks accepted from the parent chain(s).

[7] In practice, being able to include the output of a cryptographically secure hash function can be considered sufficient in the context of space requirements.

Miners participating in merged mining are required to run a full node for the respective child cryptocurrency[8]. Analogous to the normal mining process, unconfirmed transactions are assembled into blocks for both the parent and any merge-mined cryptocurrencies selected. The hash-based proof-of-work of the herein considered parent cryptocurrencies has the property that any modifications to the input, namely the block header and subsequently the entire data of the block, would invalidate the PoW with high probability. Therefore, including data, such as a hash of the to be mined child block header, anywhere in the parent block implicitly links the child block to the parent block's proof-of-work.

Miners then proceed to follow the normal mining process, looking for valid solutions to the parent's PoW puzzle. Each time a solution candidate is found, miners can perform the following actions:

- In case the PoW solution of the parent block meets the difficulty requirements of the parent cryptocurrency, miners create a regular parent block following the normal mining procedure.
- Independent thereof, if the PoW solution of the parent block meets the difficulty requirements of any of the child blockchains, the respective child block is considered valid and can be published in that child's network.

To ensure nodes in the child blockchains are able to verify the correctness of a merge-mined child block, miners must include all data relevant for validating the proof-of-work of the parent blockchain, as well as the data linking the child to the paren't PoW.

In Namecoin, for example, the parent block header and coinbase transaction are included as additional, so called *AuxPoW*, header in merge-mined Namecoin blocks. The coinbase transaction is the first transaction in a block and is used to distribute newly generated coins to miners. It also allows to store up to 96 bytes of arbitrary data in the so called *coinbase* field. During the process of merged mining Namecoin with Bitcoin, miners reference the hashes of the to-be-mined Namecoin blocks in the coinbase fields of the to-be-mined Bitcoin blocks, linking the Namecoin blocks to Bitcoin's PoW[9]. As a result, nodes in the peer-to-peer network of Namecoin can verify that the PoW for the submitted blocks was correctly performed as part of the mining process of the parent cryptocurrency, namely Bitcoin. The process of merged mining on the example of Bitcoin and Namecoin is shown in Figure 2.1, while the structure of the parent and child blocks is visualized in Figures 2.2 and 2.3.

When merged mining, each parent block can be responsible for the generation of multiple child blocks, i.e., miners may find multiple PoW solution candidates below the difficulty of the parent but meeting the requirements of the child blockchain. Furthermore, even

[8] Miners can decide to perform so called SPV mining, i.e., not verify transactions included in blocks, or simply ignore transactions at all. This, however, can be considered malicious behavior as it may damage the child blockchain.

[9] Note: this changes the final hash of the Namecoin block, as seen in the blockchain.

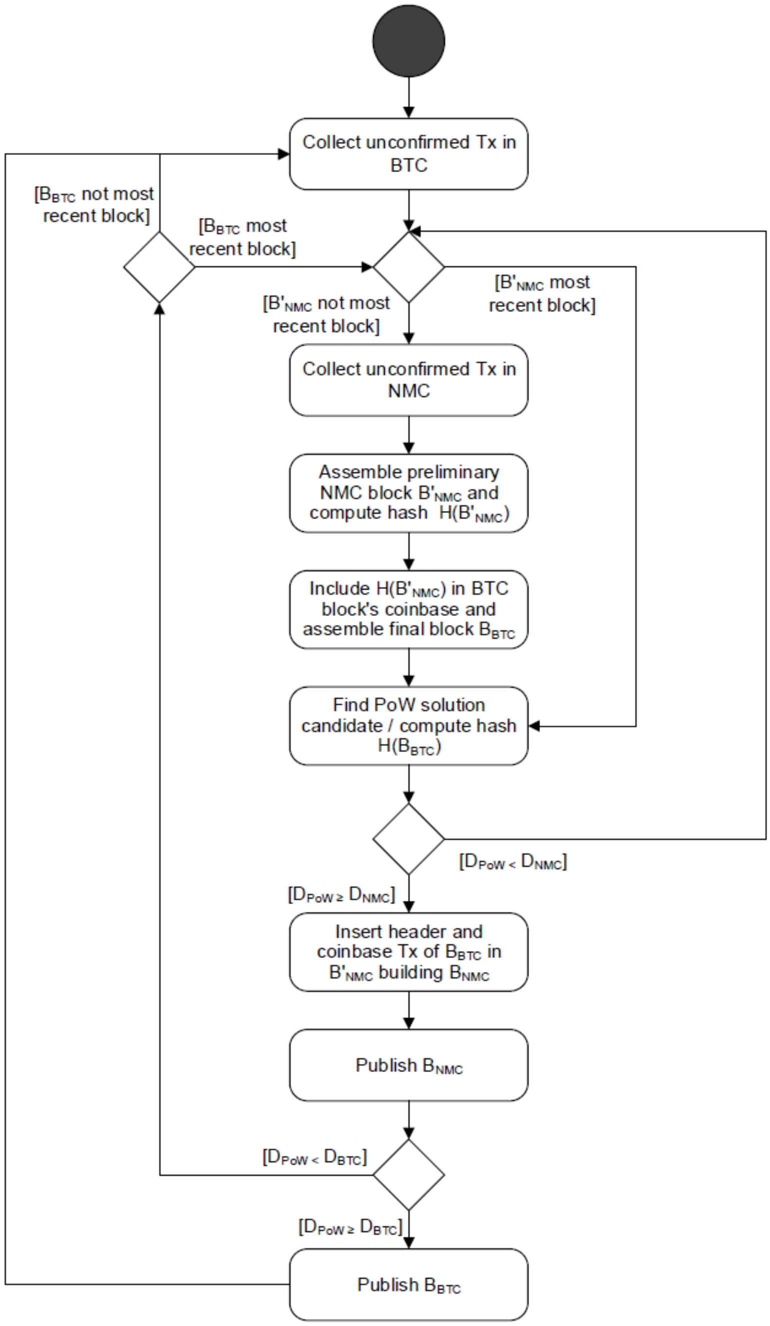

Collect unconfirmed Tx in BTC
[B_BTC not most recent block]
[B_BTC most recent block]
[B'_NMC not most recent block]
[B'_NMC most recent block]
Collect unconfirmed Tx in NMC
Assemble preliminary NMC block B'_NMC and compute hash H(B'_NMC)
Include H(B'_NMC) in BTC block's coinbase and assemble final block B_BTC
Find PoW solution candidate / compute hash H(B_BTC)
[D_PoW < D_NMC]
[D_PoW ≥ D_NMC]
Insert header and coinbase Tx of B_BTC in B'_NMC building B_NMC
Publish B_NMC
[D_PoW < D_BTC]
[D_PoW ≥ D_BTC]
Publish B_BTC

Figure 2.1: Process of merged mining as performed by miners when merged mining Namecoin with Bitcoin. Note: for simplification and based on real-world observations we assume the PoW difficulty required by Bitcoin D_{BTC} is higher than the PoW difficulty required by Namecoin D_{NMC}, i.e., $D_{BTC} > D_{NMC}$

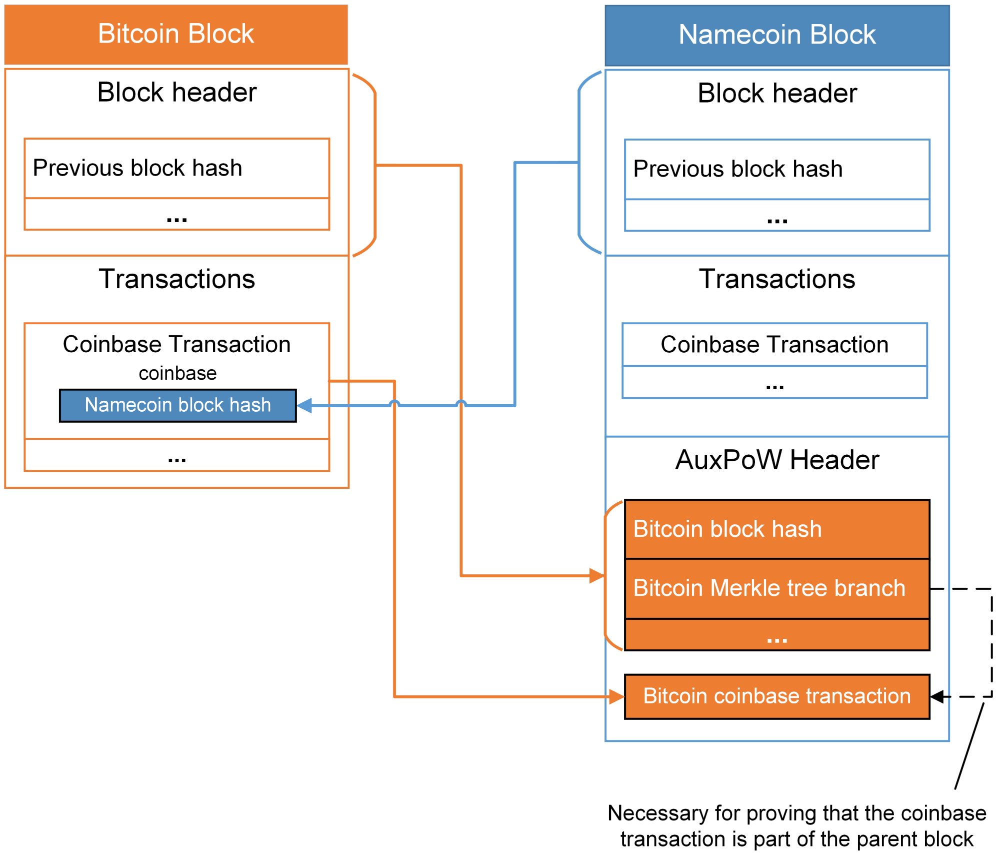


Figure 2.2: Structure of merged mined blocks in Namecoin. The block hash of the to-be-mined Namecoin block is included in the coinbase field of the Bitcoin block. Once a fitting PoW solution is found, information from the Bitcoin block header and the coinbase transaction are included in the Namecoin block. Note: the Bitcoin Merkle tree branch is necessary to verify that the included coinbase transaction was part of the respective Bitcoin block.

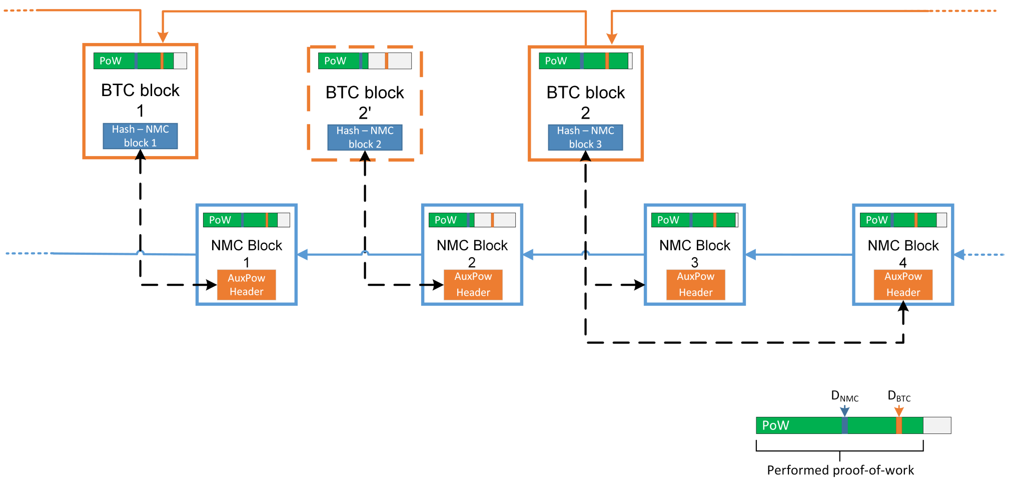


Figure 2.3: Overview of merged mining in Bitcoin with Namecoin. Assuming $D_{BTC} > D_{NMC}$, Blocks accepted in Bitcoin will be accepted in Namecoin. However, even blocks missing the difficulty target for Bitcoin, can still meet the requirements for Namecoin, as shown for *BTC Block 2'*. Furthermore, as depicted for *NMC Block 3* and *4*, a single BTC block can be referenced my multiple NMC blocks, if numerous solutions meeting D_{NMC} are found in the process of mining.

if a block mined for the parent chain does not satisfy the difficulty requirements of the parent, it can be used for generation of blocks in the child, given that the respective child difficulty is met (c.f. Figure 2.3).

As mentioned, a single parent can be used to perform merged mining on multiple child chains. This can be achieved by including the Merkle-tree root of a Merkle tree, containing the block hashes of the child blocks as leaves, in the coinbase field of the parent block. In addition, the size of the Merkle tree and the path to the position of the respective block hash must be provided. A visualization of the used Merkle tree structure is given in Figure 2.4. Thereby it is necessary to make sure that a miner cannot mine on different branches of the same child blockchain, as this conflicts with the rules of the underlying Nakamoto consensus mechanism and would make double spending attacks possible. Hence, each cryptocurrency must specify a unique ID, which can be used to derive the leave of the Merkle tree where the respective block hash must be located.

The aim of the introduction of merged mining was, on one hand, to disincentive miners of large and established cryptocurrencies like Bitcoin to switch their mining activities to emerging cryptocurrencies. As such, these miners were given the possibility to continue mining on the established cryptocurrencies, while also generating profits in merge-mined child blockchains. For small and newly created cryptocurrencies, which have not been able to accumulate a sufficiently large number of miners, merged mining provides access to the large computational capacities of established blockchains like Bitcoin. Thereby,

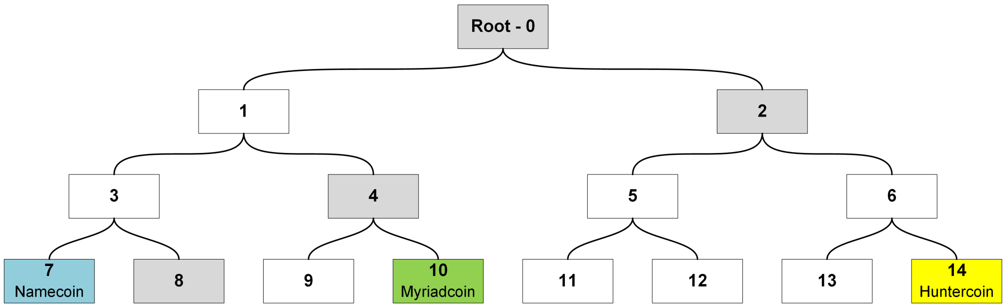


Figure 2.4: Visualization of a Merkle tree used when merge-mining multiple child cryptocurrencies in parallel. For Namecoin to be able to verify that the respective block hash is contained in the Merkle tree at position 7, hashes of the fields 8, 4 and 2 (coloured in grey) must be provided in addition to the root hash (0). Furthermore, the order in which to apply the given hashes (in our case "right", "right", "right") must be included.

the motivation for small cryptocurrencies is to make attacks on the network more costly, hence improving the security of the underlying system[10]

By implementing merged mining, child cryptocurrencies permanently bind themselves to the parent(s), becoming reliant on the respective mining community. On one hand, it becomes very difficulty for child cryptocurrencies to switch back to normal mining, without facing the threat of significant hash rate loss. In fact, to this date we are not aware of any successful attempts. On the other hand, concerns with regards to potential risks of mining power centralization have been voiced in the community, although no study has yet been conducted to verify these claims. Ideally, the permanent parent-child relation triggers no significant negative effects, apart from the child being dependent of the parent's community and developments. However, as we show in the rest of this thesis, this is not the case in reality.

Previous research related to merged mining is mostly limited to the application layer of the underlying cryptocurrencies. A short description of merged mining is provided by Kalodner et al. in an empirical study of name squatting in Namecoin [38]. Ali et al. highlight that Namecoin suffers from centralization issues linked to merged mining, but provide no detailed study on the extent of the problem, nor on merged mining in general [3]. Other descriptions of and references to merged mining can be found in [6, 26, 57, 59], whereas [8, 44] seek to employ merged mining as a component of various blockchain-based applications.

[10] In theory, miners of the small cryptocurrencies switching to merged mining move their computational resources to the parent, this way (minimally) increasing the overall mining power of the parent.

2.2.2 Weak Blocks and Subchains

The idea of *weak blocks* was initially proposed by TierNolan (Pseudonym) in 2013 [64] and later extended in Rizun's *subchain* concept [65]. Weak blocks represent otherwise valid blocks, which do not meet the target difficulty D of the underlying cryptocurrency but satisfy some lower difficulty D_{weak}, i.e. $D_{weak} < D$. Instead of being discarded, these blocks can be reused and exchanged between miners to potentially reduce transaction confirmation times.

Weak blocks form so called subchains between consecutive full blocks by referencing the previous' weak blocks block header in an additional pointer. Taking Bitcoin as example, full blocks have an interval of 10 minutes. As the difficulty target of weak blocks can be chosen arbitrarily (only requirement is that $D_{weak} < D$), the interval between such blocks can be significantly lower than that of full blocks. Taking into account that weak blocks are otherwise fully valid Bitcoin blocks, they can be used for faster (weak) transaction confirmations. The lower block intervals further can be of advantage for miners: by participating in building and validating subchains, miners can profit from being able to earlier determine diverging blockchain branches, i.e., so called *forks*. As a result, miners face a lower risk of investing computational effort on a blockchain branch, which will be later discarded as the shorter chain.

The process of mining weak blocks is similar to the normal mining process. Each time a miner finds a valid solution to the PoW of the underlying cryptocurrency, he can perform the following actions:

- If the PoW solution matches the requirements of target difficulty D, the miner creates and publishes a full block. Consequently, he starts to search for the PoW solution for the next full block.

- If the PoW solution does not fulfil the requirements of D, but meets the requirements of D_{weak}, the miner builds and publishes a weak block. Consequently, he resumes the search for the PoW solution to the current full block[11].

An exemplary visualisation of subchains created between consecutive full blocks in a blockchain is provided in Figure 2.5.

Further discussions related to the concept of weak blocks and subchains can be found in [4, 5, 48, 66, 67, 70]. Despite having seen active discussions, as of today there exists no implementation of the weak blocks concept. One possible explanation for the absence of development in this area is the lack of incentive for miners to participate in building subchains. The intrinsic rewards in form of earlier fork detection may be insufficient when

[11] Depending on the structure of the references to preceding weak blocks in the subchain, the miner may be required to adjust some parts of the block (e.g. coinbase transaction) before resuming the mining process. Otherwise, it would not be possible to build a chain of weak blocks.

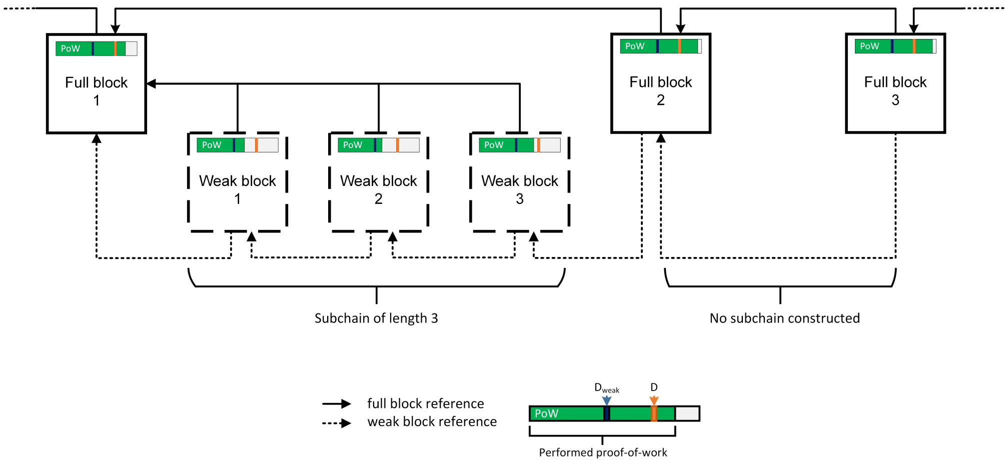


Figure 2.5: Visualization of a subchain between two full blocks. Since each weak block was initially built to become a full block, it has a reference to the previous full block. In addition, all blocks store a reference to the last block of the last subchain. If no subchain is constructed, this reference then too points to the previous full block.

put in contrast to the possible overhead for participating miners in terms of computation[12], bandwidth and maintenance.

2.2.3 P2Pool: Proof-of-Work Reusing for Share Validation

P2Pool [2] is a decentralized Bitcoin mining pool and was announced and launched in 2011. In contrast to conventional mining pools, P2Pool requires no operator to verify each miner's contribution to the mining operation of the pool. Instead, a network of peer-to-peer miner nodes is created parallel to Bitcoin and the proof-of-work of mining pool shares is reused for verification of each miner's contribution.

The key concept behind P2Pool is the so called *sharechain*. The sharechain is a fully functional blockchain, which runs in parallel to Bitcoin but maintains a significantly lower difficulty $D_{share} < D_{BTC}$, targeting a block interval of 30 seconds. Sharechain blocks are fully functional Bitcoin blocks, which meet D_{share} but not D_{BTC}, i.e., equate to shares submitted by miners to a mining pool.

Miners participating in P2Pool initially follow the normal mining process, as if solo mining. When building the block, the miner inserts his own payout address(es) in the outputs of the coinbase transaction and starts to search for solution candidates for the resulting PoW puzzle. However, in contrast to solo mining, P2Pool miners do not keep the complete block reward to themselves. Each time the miner finds a valid PoW solution where $D_{share} < D_{PoW} < D_{BTC}$, i.e., a valid share but not a full Bitcoin block, he

[12] Not proof-of-work, but verification of transactions and additional consensus rules.

publishes this block to the network of P2Pool miners. After the majority of the peers has verified that the block is valid, it is appended to the sharechain and all miners resume their search for the next Bitcoin block.

However, when building the preliminary block structure, miners now must include the payout address(es) of the previous sharechain block in the outputs of the coinbase transaction. Otherwise, P2Pool peers will discard the block as invalid and the respective miner will not be rewarded for the submitted share when a full block is found. Whenever any miner participating in the scheme finds a full Bitcoin block, he publishes it to the Bitcoin network. As a result, all miners who have submitted sharechain blocks during this round will receive a portion of the reward, directly distributed through the coinbase transaction of the block. A visualization of P2Pool's sharechain is provided in Figure 2.6

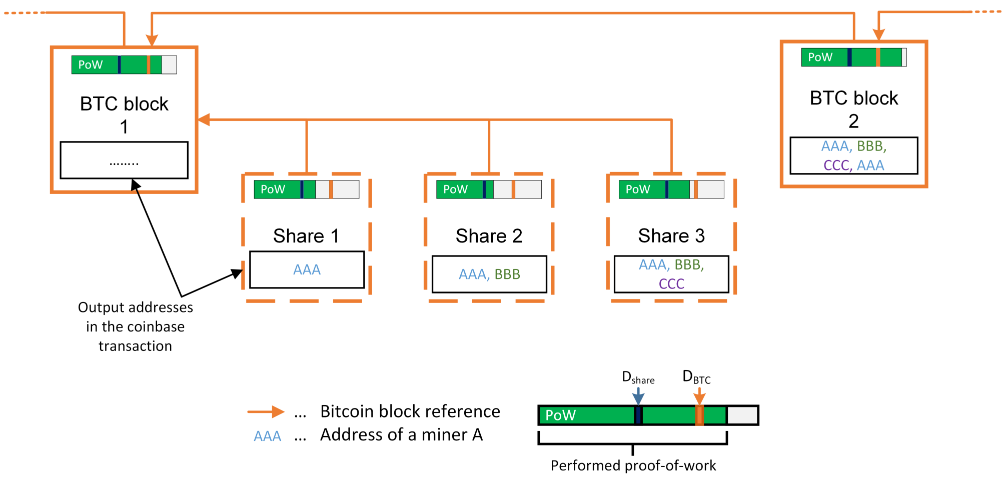


Figure 2.6: Exemplary visualization of the P2Pool sharechain and its connection to Bitcoin. Each sharechain block (denoted as *Share*) references the previous Bitcoin block. With each found share, a new address is added to the outputs of the coinbase transaction of the to-be-found Bitcoin block. Once a full block is found, all miners which have submitted shares receive rewards. In our case miner A will receive half of the generated revenue.

A mining round in P2Pool represents a sliding window 8.640 sharechain blocks, i.e, approximately three days, and the Bitcoin block rewards are split according to each miners contribution during this period. The sharechain hence only contains 8.640 block at any given point in time, always discarding the oldest share each time a new is found. Since each sharechain block references a full Bitcoin block, participating peers can verify work performed in the past by checking the Bitcoin blockchain. Since P2Pool has no central operator, no fees are charged for participation.

2.3 Proof-of-work Re-purposing

The idea of utilizing the resource intensive computations of PoW for more "useful" application has been a topic of active research in the past years. Thereby, the proof-of-work mechanism is adapted to serve more purposes than leveraging consensus in the respective blockchain/system.

A first approach was introduced by Dwork and Naor, who proposed a scheme for combating junk mail in 1992 by requiring the sender of a message to provide a proof-of-work solution as attachement [22]. Hashcash [9], first described in 1997 and further extended in 2002 [9] followed a similar motivation when introducing the idea of hash based proof-of-work. Juels et. al. apply the concept of proof-of-work as mitigation for denial-of-service attacks [36]. A further approach to re-purpose proof-of-work was introduced in the *RPOW* project by Finney in 2004 [25], aiming at allowing third parties to remotely verify what programs are running on the server hosting RPOW. The RPOW prototype was planned to be the first of a series of so called *Transparent Servers*, which publish their source code for review and are able to prove they are running the program built from the published code.

More recently, Permacoin [51] and its extension Retricoin [74] introduced the idea of re-purposing proof-of-work for data preservation. Thereby, the proof-of-work mining process as known from Bitcoin is replaced by so called *proofs-of-retrievability* [37], where miners are required prove to the network that they are storing some large amount of data. A further interesting approach is presented in Primecoin [42] where the search and discovery of long sequences of prime numbers is used as proof-of-work, although the economic applicability may appear less valuable when compared to other concepts.

Re-purposing of proof-of-work differs from proof-of-work reusing in terms of the modification of the underlying proof-of-work itself. While the approaches introduced in Section 2.2 aim at reusing proof-of-work in a self-contained manner without changing the underlying mechanism, whereas the concepts described in the previous paragraphs ultimately represent new proof-of-work mechanisms.

CHAPTER 3

Methodology

This chapter covers the methodological approach of the conducted study. In Section 3.1 we discuss the analysed set of cryptocurrencies, while Section 3.2 provides information on how the data set was collected. The novel scheme for attributing blocks to miners and mining pools introduced in this thesis is described in Section 3.3

3.1 Evaluated Cryptocurrencies

In the following paragraphs we briefly describe the cryptocurrencies, which are exemplary for merged mining and hence are considered in our study. A summary containing all relevant information is provided in Table 3.1.

Bitcoin

Bitcoin [55] represents the first and currently largest cryptocurrency. It uses SHA256[1] in its proof-of-work algorithm and maintains a target block interval of 10 minutes. Furthermore, Bitcoin represents the first cryptocurrency to be used as parent blockchain for merged mining.

Namecoin

Namecoin [58], which intends to provide a decentralized and censorship resistant alternative to the Domain Name System (DNS), was created as the first fork of Bitcoin in 2011 and represents the first alternative cryptocurrency. It uses SHA256 in its PoW algorithm and maintains a target block interval of 10 minutes. While its design is heavily based on

[1] Correctly, the notation would be *dSHA256*, as the partial pre-image attack performed as part of Bitcoin's PoW requires to find the double SHA256 hash over the given input. For better readability and simplification, we shall however use SHA256 as notation in the rest of this thesis.

Bitcoin, Namecoin extends the underlying protocol by introducing new *transaction types*, which enable the storage and management of additional information in the blockchain (e.g., DNS entries). Most important, Namecoin was the first blockchain to introduce merged mining, in this case with Bitcoin.

Litecoin

Litecoin [46] is a fork of Bitcoin launched in 2011, which replaces SHA256 with the memory-hard *Scrypt* cryptographic hash function in its PoW algorithm. Litecoin's primary aim was to counter the domination of *ASICs*, i.e., hardware devices specifically-built for high-performance SHA256 hashing operations in Bitcoin. It maintains a lower block interval than Bitcoin, namely 2.5 minutes. At the time of writing Litecoin is the largest Scrypt PoW cryptocurrency in terms of market capitalization and adoption [1]. Furthermore, Litecoin was the first cryptocurrency to be used as parent blockchain for Scrypt-based merged mining,.

Dogecoin

Dogecoin [20] initially started as a non-serious project based on an internet meme in 2013 but was able to attract and maintain a vivid community. It is roughly based on the Litecoin codebase but maintains a lower block interval of 1 minute. Dogecoin was the first cryptocurrency to introduce Scrypt-based merged mining, namely with Litecoin.

Huntercoin

Huntercoin [32] was launched in 2014 and is the first cryptocurrency to build a game on top of a blockchain, aiming to support *human mining*. Furthermore, it is the first of a new generation of so called *multi-PoW* cryptocurrencies, which combine multiple proof-of-work algorithms in a single system. The concept of multi-PoW aims to provide resistance to mining centralization, as a potential adversary would now be required to accumulate hardware for all supported PoW algorithms. Huntercoin uses SHA256 and Scrypt as PoW algorithms and maintains a block time of 1 minute. It further represents the first so called *multi-merge-mined* cryptocurrency, as it allows merged mining with both PoW algorithms in parallel, i.e. with Bitcoin and Litecoin.

Myriadcoin

Myriadcoin [54], also launched in 2014, extends the multi-PoW idea of Huntercoin, introducing support for five different PoW algorithms, including SHA256 and Scrypt. Furthermore, it uses guards to prevent a single PoW algorithm from dominating the generation of blocks: no more than five consecutive blocks are allowed to be created using the same algorithm. Similar to Huntercoin, Myriadcoin supports both SHA256 and Scrypt-based merged mining.

Table 3.1: Summary of cryptocurrencies studied in this thesis. Note: the "merged mining role" describes the role of the cryptocurrency in the analysis of this paper. In theory, each of these could also act as parent for some other blockchains.

Blockchain	**PoW Algorithm(s)**	**Merged Mining Role**	**Block Interval (in min)**	**Launched**
Bitcoin	SHA256	Parent	10	2009
Namecoin	SHA256	Child	10	2011
Litecoin	Scrypt	Parent	2.5	2011
Dogecoin	Scryot	Child	1	2013
Huntercoin	SHA256, Scrypt	Child	1	2014
Myriadcoin	SHA256, Scrypt, Myr-Groestl, Skein, Yescrypt	Child	5	2014

3.2 Dataset Collection

For our analysis we rely on the open and publicly-accessible ledgers (i.e., blockchains) of the examined cryptocurrencies, as they represent the most reliable source of information with regards to historical data[2].

To extract and store data in an easily processable way, we re-implement the Namecoin extraction tool initially introduced in [80]. The new *blockchain data mining tool*, implemented in Java using the Spring-Boot framework [79] and Maven [47] as build tool, is constructed to be generically extendible to support Bitcoin-like cryptocurrencies. To collect data, the tool repeatedly queries the REST and JSON-RPC APIs of the reference client implementations of the analyed cryptocurrencies and maps the retrieved information to a relational scheme. The tool has been tested and deployed with PostgreSQL [53] versions 9.3-9.5, but is compatible with other major providers of relational databases, such as MySQL and OracleDB.

The results presented in the rest of this paper are based on data collected from Bitcoin, Litecoin, Namecoin, Dogecoin, Huntercoin and Myriadcoin up to a cut-off date set to June 18, 2017 23:59:59 (UTC). The resulting block heights are provided in Table 3.2 To perform evaluations of the collected data, we use Python 3.x in combination with Jupyter Notebook [43].

[2] While some public APIs are available for Bitcoin (e.g., `http://blockchain.info/`), online sources the other cryptocurrencies are scarce and not well-maintained.

Table 3.2: Cut-off block heights of the cryptocurrencies analysed in this thesis.

Blockchain	Blockheight	Date/Time (UTC)
Bitcoin	471,892	Sun, 18 Jun 2017 23:59:30
Namecoin	347,175	Sun, 18 Jun 2017 23:59:02
Litecoin	1,224,533	Sun, 18 Jun 2017 23:54:16
Dogecoin	1,763,524	Sun, 18 Jun 2017 23:57:11
Huntercoin	1,788,998	Sun, 18 Jun 2017 23:59:48
Myriadcoin	2,089,974	Sun, 18 Jun 2017 23:55:51

3.3 Block Attribution Scheme

A key element for the investigation of mining power centralization issues is a correct attribution of blocks to the original miners. Hence, we devise an attribution scheme using publicly-available information contained in the *coinbase transactions* of both the parent and child blockchains as indicators. The *coinbase transaction* thereby represents the first transaction in a block and creates new currency units as reward for its miner. In our analysis, we rely on the following pieces of information:

Reward payout addresses

Every coinbase transaction must have at least one output address, which receives the newly generated units of the underlying cryptocurrency. Assuming miners act rationally and profit-oriented, they are expected to specify one or more of their own addresses as output of this transaction. Otherwise, they face the risk of loosing reimbursement for their invested computational effort. Hence, the reward payout addresses of blocks can be used as strong indicator for mapping blocks to miners in the attribution scheme.

Coinbase signatures (markers)

Miners and especially mining pools often utilize the `coinbase` field of the coinbase transaction to publicly claim the creation of the respective block, by inserting their so-called *block-* or *coinbase signature*, e.g. "Mined by AntPool". As the latter represents a human-readable string indicating the pool name or an abbreviation thereof, rather than a cryptographically-strong signature, we hereafter refer to this piece of information as *marker*. We note, however, that a miner could decide to try to impersonate another miner or mining pool by using a fake marker. Further discussion on this is provided in Section 5.3.2.

At the time of writing, there exists no official global registry for markers or reward payout addresses of miners or mining pools[3]. Therefore, this information must be collected by analysis of publicly-available records including but not limited to websites of mining pools

[3] To the best of our knowledge, the most detailed list of Bitcoin mining pools maintained on

and discussion forums, as well as direct contacts with pool operators. As an outcome of this process, we are able to compile a list of block attribution indicators for 95 miners and mining pools, which operated in the observed cryptocurrencies.

3.3.1 Linking and Clustering of Addresses

Merge-mined blocks can contain up to four attribution indicators: the coinbase marker and reward payout addresses of the child chain, as well as the coinbase marker and reward payout addresses of the parent chain, which are stored in the so called *AuxPoW* header[4]. This allows to establish connections between reward payout addresses across multiple cryptocurrencies and to detect if miners switch between multiple addresses. Hence, reward payout addresses appearing in parent and child coinbase transactions of all blocks are checked for intersections. More specific: an address of the parent chain appearing in the coinbase of the AuxPow header allows to link it to the child chain address used in the coinbase transaction of the block. The child chain address in turn can appear in blocks together with other parent chain addresses, creating more links, and so on, thereby creating address *clusters* across multiple cryptocurrencies. Thereby, each additional merge-mined cryptocurrency allows to increase the set of addresses attributed to an address cluster of a miner, mining pool or set of miners.

For example, assume a miner or mining pool engages in merged mining of Namecoin with Bitcoin. Further, assume this miner declares his Bitcoin address as *0x1BBB* and uses *0xMAAA* as Namecoin payout address (can be publicly declared or derived from mining activity in Namecoin). Hence, the Bitcoin address *0x1BBB* will appear in the coinbase transaction outputs in the AuxPoW header of Namecoin blocks merge-mined by this miner. A visualization of this example is provided in Figure 3.1.

Now assume this miner maintains a second Bitcoin address *0x1CCC* but does not declare it publicly, i.e., mining activity using this address would remain undisclosed in Bitcoin. If the miner now decides to also use this address for merged mining activities, wiring payouts to the same Namecoin address *0xMAAA*, a publicly visible and provable link is established between those two Bitcoin addresses (*0x1BBB* and *0x1CCC*) in the Namecoin blockchain. Similar, assume this miner maintains a second undeclared Namecoin address *0xMDDD* and uses it for claiming block rewards when merged mining with Bitcoin. If this Namecoin address appears significantly often in blocks which contain the publicly known Bitcoin address *0x1BBB* in the AuxPow header, it can be provably linked to the Bitcoin miner. We note that this scheme for linking and creating address *clusters* can be successful even if no single transaction was made between *0x1BBB* and *0x1CCC* in Bitcoin, or *0xMAAA* and *0xMDDD* in Namecoin.

a voluntary basis by some community members can be found here: `github.com/blockchain/Blockchain-Known-Pools/blob/master/pools.json`

[4] Additional header in merge-mined blocks, used to verify the PoW performed in the parent chain.

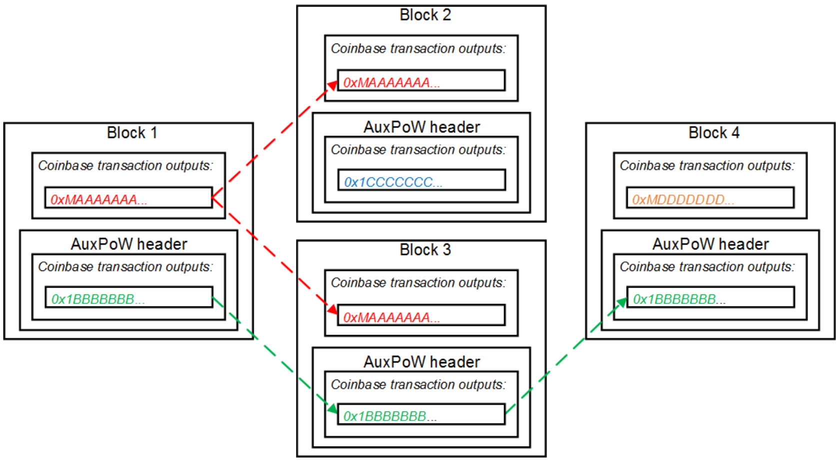

Block 1
Coinbase transaction outputs:
0xMAAAAAAA...
AuxPoW header
Coinbase transaction outputs:
0x1BBBBBBB...
Block 2
Coinbase transaction outputs:
0xMAAAAAAA...
AuxPoW header
Coinbase transaction outputs:
0x1CCCCCCC...
Block 3
Coinbase transaction outputs:
0xMAAAAAAA...
AuxPoW header
Coinbase transaction outputs:
0x1BBBBBBB...
Block 4
Coinbase transaction outputs:
0xMDDDDDDD...
AuxPoW header
Coinbase transaction outputs:
0x1BBBBBBB...

Figure 3.1: Simplified visualization of intersection detection between parent (*0x1BBB* and *0x1CCC*) and child chain (*0xMAAA* and *0xMDDD*) reward payout addresses. *0x1CCC* and *0xMDDD* represent newly identified address of the miner/mining pool associated with addresses *0x1BBB* and *0xMAAA*.

3.3.2 Attributing Blocks to Miners

A block is considered attributed to a miner if one of his reward payout addresses or markers appears in the respective fields of the coinbase transaction. However, a miner is technically allowed to use this first transaction to immediately split the block rewards to multiple outputs, this way also potentially obfuscating his identity. In such cases, it is not easily possible to determine the miner of a block, unless a known coinbase marker is used or all addresses appearing in the outputs of the coinbase transaction are associated with the same miner or mining pool. If this is the case, the block is marked as *non-attributable*. A visualization of the scheme for parent blockchains is provided in Figure 3.2 and in Figure 3.3 for merge-mined child blockchains. Payout addresses appearing often as single output in mined blocks but which cannot be linked to an identified miner or mining pool are denoted as *other unknown miners*.

However, for a permissionless proof-of-work cryptocurrency, where participants are not obliged to disclose their activity, it is not feasible for a third party to fully reconstruct a miner's history of action retroactively. Furthermore, miners may actively try to hide their identity by avoiding the reuse of payout addresses, not using any markers or using markers associated with other identities. Hence, it is not possible to identify all miners and mining pools with 100% accuracy by relying only on the information present in the public ledger if miners want to stay anonymous.

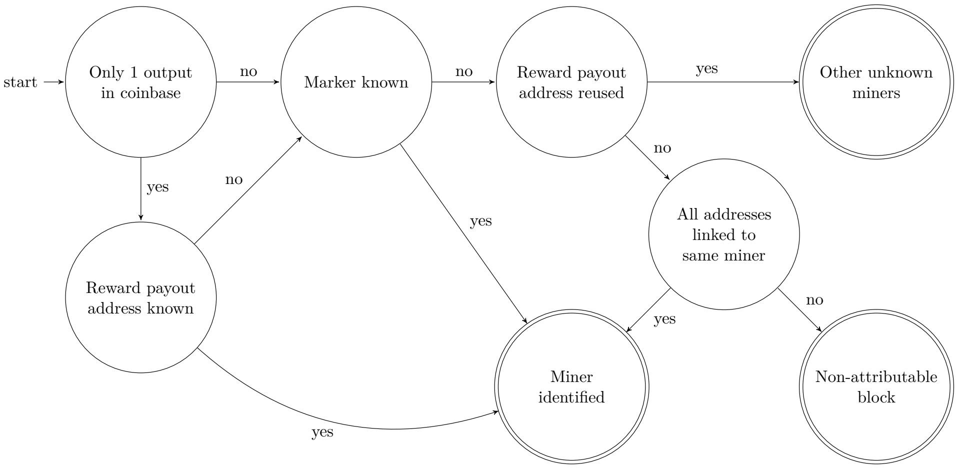

start
Only 1 output in coinbase
no
Marker known
no
Reward payout address reused
yes
Other unknown miners
yes
no
no
Reward payout address known
yes
All addresses linked to same miner
no
yes
Miner identified
Non-attributable block
yes

Figure 3.2: Block attribution scheme for parent blockchains utilizing coinbase markers and payout addresses for identification. Applied to Bitcoin and Litecoin in our study.

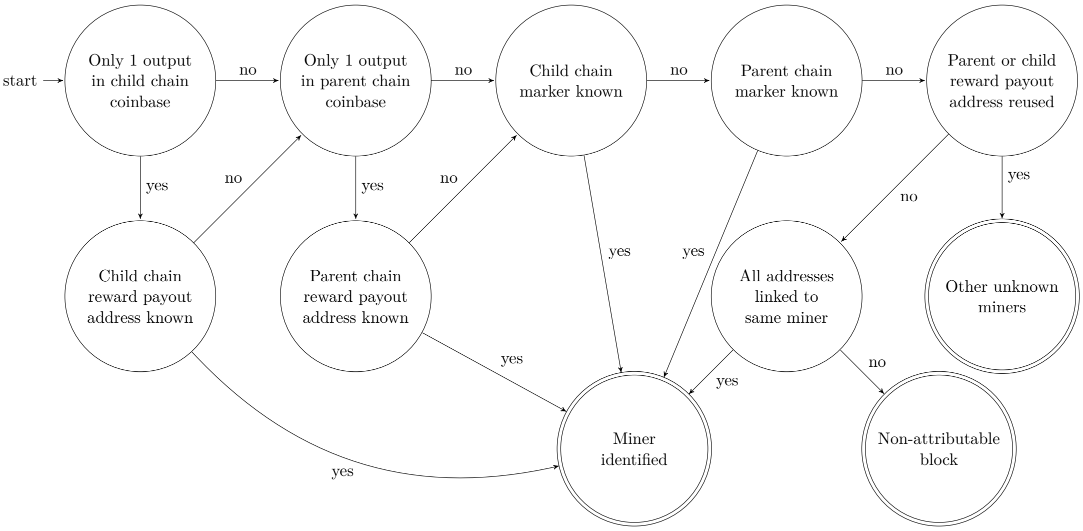

start
Only 1 output in child chain coinbase
no
Only 1 output in parent chain coinbase
no
Child chain marker known
no
Parent chain marker known
no
Parent or child reward payout address reused
yes
no
yes
no
no
yes
Child chain reward payout address known
Parent chain reward payout address known
yes
yes
All addresses linked to same miner
Other unknown miners
yes
no
yes
Miner identified
Non-attributable block
yes

Figure 3.3: Block attribution scheme for merge-mined blockchains utilizing coinbase markers and payout addresses of both parent and child coinbase transactions for identification. Applied to Namecoin, Dogecoin, Huntercoin and Myriadcoin in our study.

CHAPTER 4

Results

In this chapter we present the results of our analysis of merged mining and provide evidence for mining power centralization issues in the implementing cryptocurrencies. The rest of this chapter is structured as follows. Section 4.1 discussed the popularity and rate of adoption of merged mining. In Section 4.2 we provide insight on the effects merged mining had on the difficulty of the child cryptocurrencies. We then present the results of our block attribution scheme in Section 4.3 and show how merged mining impacts distribution of mining power in child cryptocurrencies in Section 4.4. Finally, we discuss the strong mining power fluctuation in merge-mined cryptocurrencies in Section 4.5.

4.1 Degree of Adoption

Merged mining was introduced at block 19,200 in Namecoin (Oct. 2011), 11,163 in Huntercoin (Feb. 2014), 371,337 in Dogecoin (Sept. 2014) and 1,402,791 in Myriadcoin (Sept. 2015). The developers of Namecoin, Dogecoin and Huntercoin also disabled normal mining in the official clients at introduction. Hence, from that point forward over 99% of the blocks have been created through the process of merged mining in these cryptocurrencies. Table 4.1 shows the total distribution of normal and merge-mined blocks.

Table 4.1: Merge-mined blocks in examined cryptocurrencies.

Blockchain	Normal	Merge-mined	% of Total
Huntercoin	15,083	1,773,916	99.2
Namecoin	19,330	327,846	94.4
Dogecoin	373,927	1,389,553	78.8
Myriadcoin	1,789,994	299,981	14.4

An indicator for the amount of cryptocurrencies allowing merged mining can be derived from the structure of merge-mined blocks. Recall: since multiple cryptocurrencies can be merge-mined in parallel with the same parent chain, it is possible to include the root hash of a Merkle tree, containing the block hashes of the selected child chains as leaves, in the AuxPoW header instead of the block hash of a single child cryptocurrency (c.f. Figure 2.4). Thereby, apart from the Merkle tree branch each block contains a `MerkleSize` field, which determines the numbers of leaves in the Merkle tree and thus acts as an upper bound for the number of child blockchains the miner is potentially merged mining in parallel.

An analysis of the used Merkle tree structures in merge-mined cryptocurrencies shows a general trend towards Scrypt-based blockchains with regards to the number of child chains possibly merge-mined in parallel. For example in Namecoin, 53.4% of the merge-mined blocks were designed to be applicable solely for Namecoin, while 74.5% of the merge-mined blocks in Dogecoin have *potentially* been used in other Scrypt-based altcoins as well. An overview of the distribution of the upper-bounds of parallel merge-mined blocks in the studied cryptocurrencies is provided in Table 4.2. We note, however, that the Merkle tree structure can be chosen freely and hence these numbers can only act as rough estimates.

Table 4.2: Distribution of blocks potentially merge-mined with other child chains in parallel (upper bounds).

	1 (only self)	2	4	8	16	32	64	$\geq$ 128
Namecoin	53.40	14.00	19.90	3.11	9.48	0.01	0.00	0.10
Huntercoin (SHA256)	0.31	21.61	6.96	10.70	41.30	8.87	2.84	7.41
Myriadcoin (SHA256)	6.76	2.86	3.67	27.90	25.40	17.00	4.44	11.87
Dogecoin	25.50	0.84	1.26	25.20	3.30	0.90	43.0	0.00
Huntercoin (Scrypt)	0.00	2.37	5.23	10.80	0.00	0.00	81.6	0.00
Myriadcoin(Scrypt)	0.14	0.00	0.02	0.16	61.30	0.20	38.1	0.08

4.2 Effects on PoW Difficulty

The main objective of merged mining is to attract more miners and hence increase the difficulty of the child blockchain [59]. By extracting the information on the PoW difficulty encoded in each block header, we are able to confirm merged mining indeed has a positive effect in this respect.

Figure 4.1 visualizes the development of the SHA256 PoW difficulty in Bitcoin compared to Namecoin, Huntercoin and Myriadcoin on a logarithmic scale. After the introduction of merged mining, all child cryptocurrenties experienced a rapid increase of the PoW difficulty. Furthermore, the behavior of Bitcoin's difficulty is, to some extent, mirrored to the merge-mined cryptocurrencies. For example, between January 2012 and April 2013 the difficulty remained stable in both Bitcoin and Namecoin, until an upward trend occurred in May 2013. The latter coincides with the wide deployment of specialized hardware dedicated to mining (ASICs) [76]. Similar observations are made for Litecoin and Scrypt merge-mined cryptocurrencies, as shown in Figure 4.2.

We note that after a period of growth, the difficulties of both SHA256 and Scrypt PoW algorithms in Huntercoin experienced a significant drop in mid 2016. Closer analysis shows this development was caused by F2Pool, the largest mining pool in Huntercoin at that time, ceasing its merged mining operations in this cryptocurrency. A further in interesting observation can be made in the case of Myriadcoin: the Scrypt PoW difficulty of this multi-merge-mined cryptocurrency recently exceeded that of Litecoin, one of its parent blockchains, by 31,85%. This can be explained by both the higher block interval in Myriadcoin (5 minutes in contrast to 2.5 minutes in Litecoin) and the increase of participating miners.

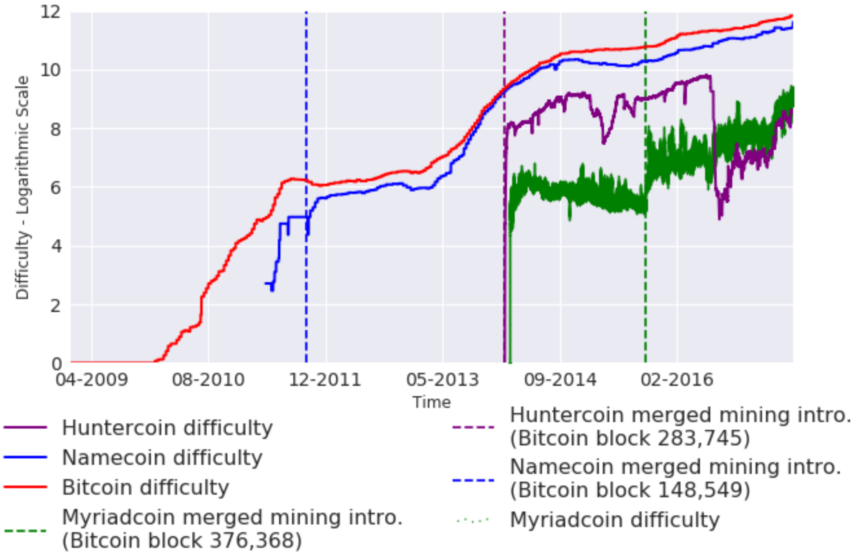


Figure 4.1: Difficulty development in Bitcoin compared to SHA256 merge-mined cryptocurrencies over time on a logarithmic scale (since the launch of Bitcoin).

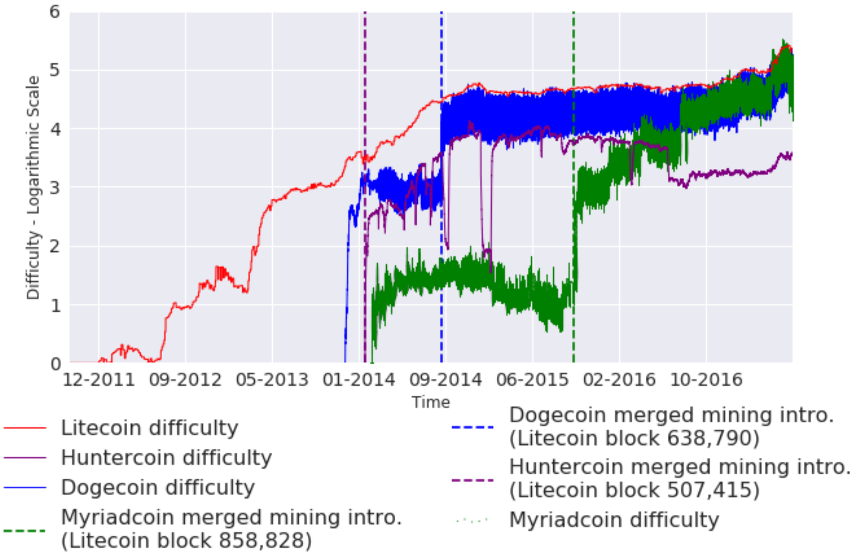


Figure 4.2: Difficulty development in Litecoin compared to Scrypt merge-mined cryptocurrencies over time on a logarithmic scale (since the launch of Litecoin).

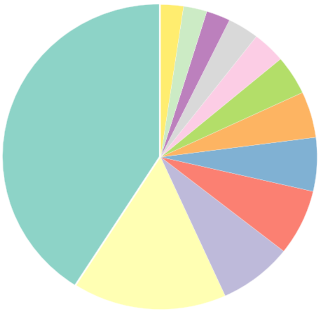
4.3 Impacts on Mining Power Distribution

In order to investigate the connection of merged mining and mining power centralization, we apply the attribution scheme described in Section 3.3 on the dataset of evaluated cryptocurrencies. Thereby, we consider a block successfully mapped, if we can attribute it to either a known mining pool, or a reused reward payout address. As a result, we are able to map the following percentage of blocks within the respective cryptocurrency: 59.1% for Bitcoin, 88.5% for Namecoin, 73.2% for Litecoin, 99.5% for Dogecoin, 82.7% for Huntercoin and 87.2% for Myriadcoin.

The low attribution success rate for Bitcoin can be explained by taking into consideration its early mining landscape, where blocks were primarily mined by individuals. At that time it was generally considered best practice not to reuse reward payout addresses and the official client exhibited this behavior. The utilization of markers and reuse of payout addresses became observable only once miners started to join forces by forming mining pools in late 2011. Similar observations can be made in the older cryptocurrencies Namecoin and Litecoin, albeit at a smaller scale. By the time Dogecoin, Huntercoin and Myriadcoin were launched, mining had already become competitive and most operations were run by mining pools, making the attribution of blocks easier.

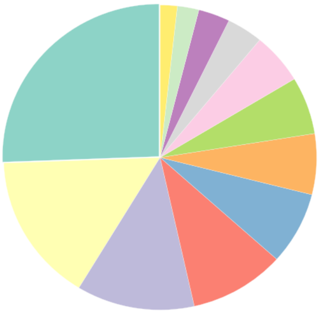
The attribution results, summarized in Figures 4.3 - 4.8, suggest that a small set of mining pools are able to control significant portions of the overall mining power across multiple cryptocurrencies. While in some cases this is explained by their long-term commitment to mining on the respective chain, pools like *GHash.IO*, *BW Pool* and *F2Pool* appear to have enough capacity to concurrently conduct competitive mining operations in both Bitcoin and Litecoin, i.e., on different PoW algorithms. In fact, F2Pool, which represents one of the largest mining pools across both SHA256 and Scrypt PoW cryptocurrencies, was able to accumulate block shares exceeding the security guarantees of the Nakamoto consensus protocol.

However, not all miners and mining pools currently participate in merged mining. A possible explanation is the economies of scale attributed to merged mining [59]. Since no additional computational effort is required for the PoW, the costs of merged mining, namely bandwidth, storage and validation of blocks/transactions, are the same for all miners, regardless of their mining power. In particular smaller mining operations may face the situation that their additional expenditures for merge-mining another cryptocurrency exceed the expected rewards.



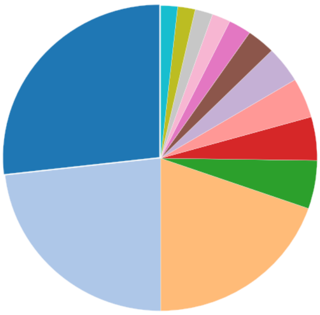
	Pool	**Blocks**	**(%)**
🟨	Smaller pools (share <1.5%)	74,753	15.8
🟪	F2Pool	35,955	7.62
🟥	BTC Guild	32,932	6.98
🟦	AntPool	26,884	5.70
🟧	GHash.IO	23,063	4.89
🟩	SlushPool	19,650	4.16
🟪	BitFury	16,070	3.41
⬜	BTCC	15,228	3.23
🟪	Other unknown miners	11,706	2.48
🟩	Eligius	11,424	2.42
🟨	BW Pool	11,075	2.35
	Attributed (total)	278,740	59.1
🟦	Non-attributable blocks	193,151	40.9

Figure 4.3: Bitcoin block attribution



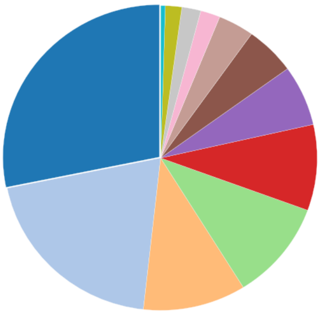
	Pool	**Blocks**	**(%)**
🟦	F2Pool	88,795	25.6
🟨	BTC Guild	54,623	15.7
🟥	GHash.IO	34,239	9.86
🟦	SlushPool	26,726	7.70
🟧	Smaller pools (share <1.5%)	24,832	7.15
🟩	Eligius	21,144	6.09
🟪	BitMinter	18,788	5.41
⬜	EclipseMC	12,954	3.73
🟪	BTCC	11,298	3.25
🟩	ViaBTC	7,734	2.23
🟨	N3aNrkyTKY...	6,027	1.74
	Attributed (total)	307,160	88.5
🟪	Non-attributable blocks	39,927	11.5

Figure 4.4: Namecoin block attribution



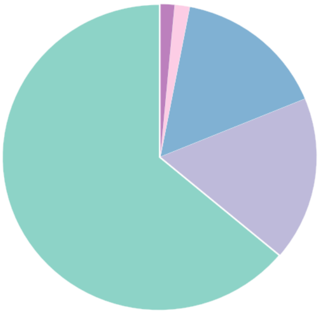
	Pool	Blocks	(%)
🟦	Smaller pools (share <1.5%)	284,339	23.2
🟧	F2Pool	240,691	19.7
🟩	LTm3aN5CbZ...	62,623	5.11
🟥	Clevermining	56,340	4.60
🟥	Other unknown miners	51,671	4.22
🟪	BW Pool	47,229	3.86
🟫	litecoinpool.org.	35,806	2.92
🟪	LTC1BTC/LTC.BTC.TOP	28,627	2.34
🟪	LTZaRkmkTJ...	23,342	1.91
⬜	GHash.IO	22,435	1.83
🟨	LiteGuardian	22,148	1.81
🟦	Give Me Coins	21,299	1.74
	Attributed (total)	896,550	73.2
🟦	Non-attributable blocks	327,984	26.8

Figure 4.5: Litecoin block attribution



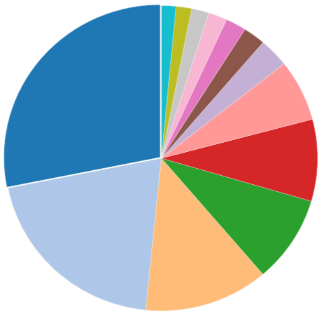
	Pool	Blocks	(%)
🟦	F2Pool	497,013	28.2
🟦	Other unknown miners	353,671	20.1
🟧	Clevermining	187,376	10.6
🟩	Smaller pools (share <1.5%)	186,348	10.6
🟥	Litecoin pool using LTm3aN5CbZ2Ns34...	160,644	9.11
🟪	litecoinpool.org.	113,283	6.42
🟫	BW Pool	91,265	5.18
🟫	LTC1BTC/LTC.BTC.TOP	65,228	3.70
🟪	yihaochi.com	35,745	2.03
⬜	Coinotron	34,694	1.97
🟨	GHash.IO	29,814	1.69
	Attributed (total)	1,755,081	99.5
🟦	Non-attributable blocks	8,443	0.5

Figure 4.6: Dogecoin block attribution



	Pool	Blocks	(%)
🟩	F2Pool	1,142,821	63.9
🟦	litecoinpool.org.	282,136	15.8
🟪	HaoBTC	27,974	1.56
🟪	Smaller pools (share < 1.5%)	26,057	1.46
	Attributed (total)	1,478,988	82.7
🟪	Non-attributable blocks	310,010	17.3

Figure 4.7: Huntercoin block attribution



	Pool	Blocks	(%)
🟦	Smaller pools (share <1.5%)	587,986	28.1
🟦	Other unknown miners	423,684	20.3
🟩	Nonce-pool	192,193	9.20
🟥	MiningPoolHub	181,168	8.67
🟥	Zpool	135,876	6.50
🟪	MJv9fLd7Qj...	64,720	3.10
🟫	LTC1BTC/LTC.BTC.TOP	48,132	2.30
🟪	Multipool	44,510	2.13
🟪	MWQVvPypce...	40,281	1.93
⬜	GHash.IO	37,916	1.81
🟨	Wafflepool	33,605	1.61
🟦	Nut2Pools	31,359	1.50
	Attributed (total)	1,821,430	87.2
🟧	Non-attributable blocks	268,544	12.8

Figure 4.8: Myriadcoin block attribution

4.4 Resulting Mining Power Centralization Issues

The number of blocks found by a miner over a certain period indicate his actual hash rate (i.e., the mining power) during this period. Hence, we use the number of blocks generated by the largest miner or mining pool per day as an approximation for measuring the centralization of mining power. Thereby, we set the observation period to 24 hours to avoid extreme variance caused by lucky/unlucky streaks of miners since the time between found blocks is exponentially distributed, while still achieving accurate results.

Our findings are visualized as heatmaps in Figures 4.9 - 4.11. Therein, each bar (column) represents the number of blocks mined by the largest entity on that day. We use the thresholds described in Section 2.1.2 as centralization indicators. If exceeded, the latter are known to introduce potential threats on the decentralization and security level of PoW blockchains:

- *Below 25%* (green) - Highest share is below the pessimistic threshold, i.e., no miner or mining pool is able to accumulate significant portions of the overall mining power.
- *Greater 25%* (yellow) - Highest share controlled by a single miner or mining pool lies between 25% and 33.33%, i.e., above the pessimistic threshold for the security of the underlying consensus mechanism.
- *Greater 33.33%* (orange) - Highest share controlled by a single miner or mining pool lies between 33.33% and 50%. Significantly high success rates of attacks such as selfish mining.
- *Greater 50%* (red) - A single entity controls the majority of mining power.

In Bitcoin no single miner or mining pool has been able to aggregate and maintain more than 50% of the overall mining power for an extended period, since blocks became attributable [1]. However, the situation is quite different in Namecoin: here, *F2Pool* reached and maintained a majority of the mining power for prolonged periods, at times being responsible for over 80% of the generated block per day.

Litecoin, despite being the largest Scrypt PoW blockchain, has experienced slight centralization since mid-2014, among others caused by increasing mining power of *Clevermining* and lately *F2Pool*. Through merged mining, this situation is reflected and amplified in Dogecoin: *F2Pool* was responsible for generating more than 33% of the blocks per day for significant periods, even exceeding the 50% threshold around the end of 2016.

The effects of introducing merged mining have played out differently in the two studied multi-Pow cryptocurrencies Huntercoin and Myriadcoin. Huntercoin was instantly dominated by *F2Pool* and remained in this state until mid-2016. Myriadcoin, one the other hand, appears to have experienced only a moderate impact, continuing to maintain a more or less balanced distribution of mining power. However, we note that so far none of the large mining pools, active in other merge-mined chains, have been observed to operate in Myriadcoin. In theory, a mining pool would have to run mining operations on at least three of the five proof-of-work algorithms implemented by Myriadcoin, to be able to control the generation of the majority of blocks.

An overview of these observations is provided in Table 4.3. Summarizing, since not all miners and mining pools engage in merged mining, a small set of pools is able to accumulate significant mining power shares in merge-mined cryptocurrencies. This in turn leads to centralization in systems, where security guarantees rely on decentralization and an honest majority of participants.

Table 4.3: Distribution of overall percentage of days below/above the centralization indicator thresholds.

Blockchain	**< 25%**	**≥ 25%**	**>33.33%**	**>50%**
Bitcoin	75.7	24.3	5.43	0.03
Namecoin	11.7	88.3	66.6	30.5
Litecoin	45.0	55.0	35.9	0.75
Dogecoin	16.3	83.7	60.7	2.45
Huntercoin	1.53	98.5	96.1	81.0
Myriadcoin	87.7	12.3	6.20	0.2

[1] It is in the realm of possibility that in the early days of Bitcoin individual miners, such as Satoshi Nakamoto himself have controlled large shares of the overall mining power.

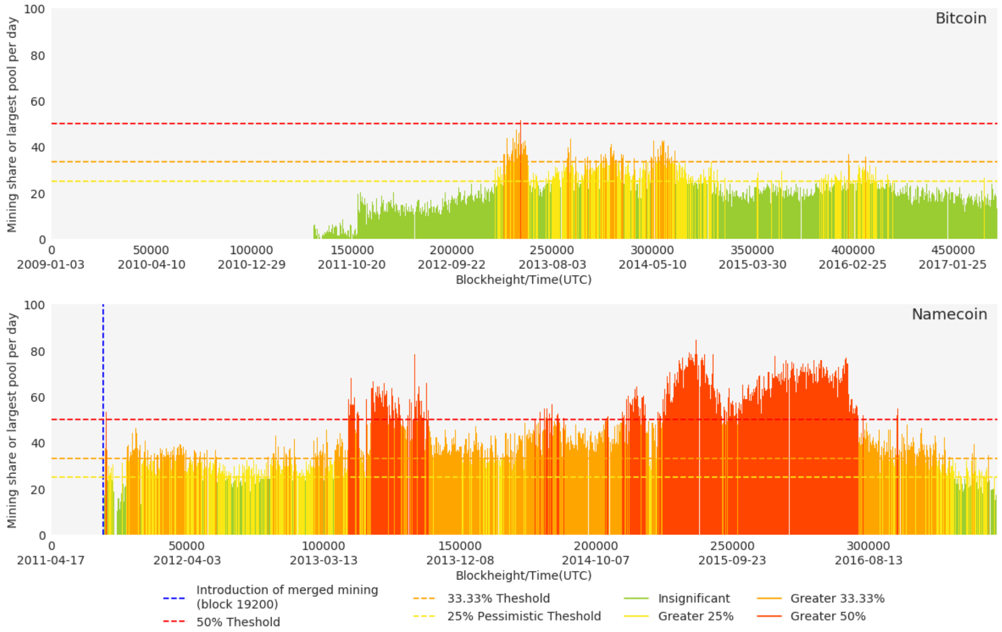


Figure 4.9: Block share of largest miner / mining pool per day for Bitcoin (144 blocks) and Namecoin (144 blocks) since launch of the respective cryptocurrency.

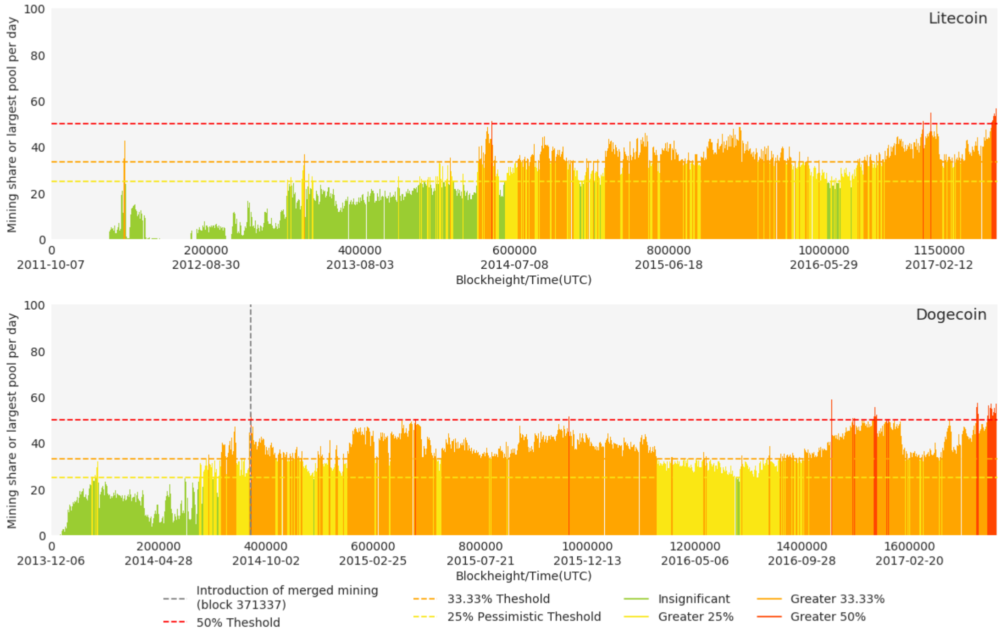


Figure 4.10: Block share of largest miner / mining pool per day for Litecoin (576 blocks) and Dogecoin (1,440 blocks) since launch of the respective cryptocurrency.

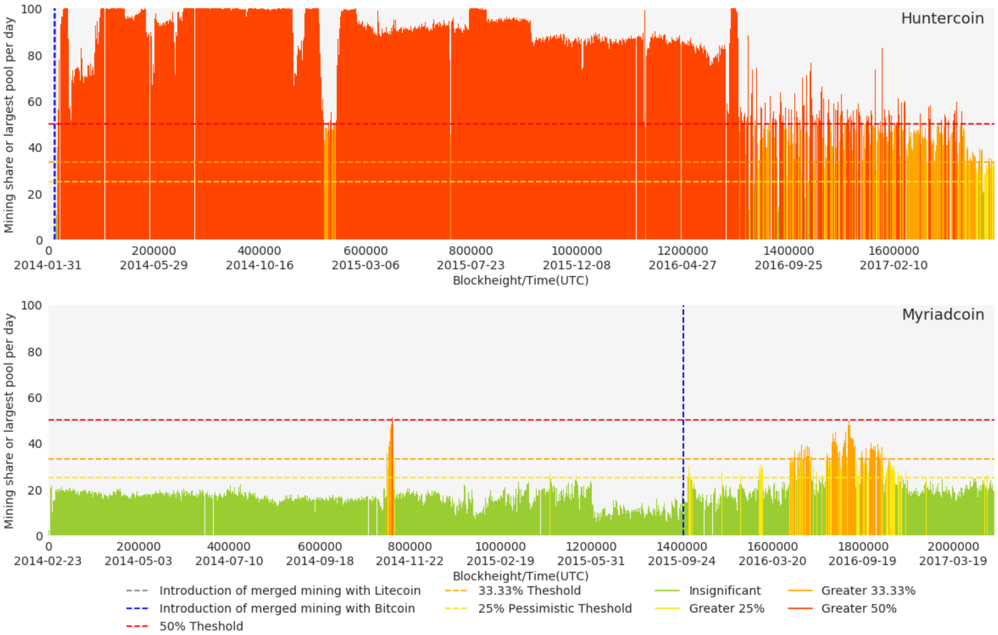


Figure 4.11: Block share of largest miner / mining pool per day for Huntercoin (1,440 blocks) and Myriadcoin (1,440 blocks) since launch of the respective cryptocurrency.

4.5 Mining power fluctuation

The operation of a mining pool requires extensive coordination effort in terms of recruiting miners or purchasing and installing the necessary infrastructure. Hence, it usually takes time until a mining pool is able to accumulate significant mining power shares. Merged mining, however, requires only minimal effort and can be described as a "software switch". Consequently, the observable high fluctuations of mining power in merge-mined cryptocurrencies may be attributed to mining pools being able to easily start or end their operation without major preparations. Figures 4.12 provides a detailed visualization of the development of mining power shares over time in Bitcoin compared to Namecoin. In contrast to Bitcoin, where the mining power shares of mining pools are mostly constant or increasing/decreasing slowly, significant fluctuation of the mining power distribution can be observed in Namecoin, e.g. around block 300,000. Detailed visualizations for the other studied cryptocurrencies are provided in the Appendix.

A further interesting observation is the increase of non-attributable blocks occurring simultaneously to drops of mined blocks that are attributable to large mining pools. Such behavior is observed in Namecoin (cf. Figure 4.12 approximately at Namecoin block 250,000) [2]. Further analysis and investigation into such events is necessary to rule out

[2] The interested reader can refer to similar visualizations for Litecoin, Dogecoin, Huntercoin and Myriadcoin in the Appendix.

that these are attempts of pools to conceal their total mining power when operating near or beyond the security guarantees offered by Nakamoto consensus

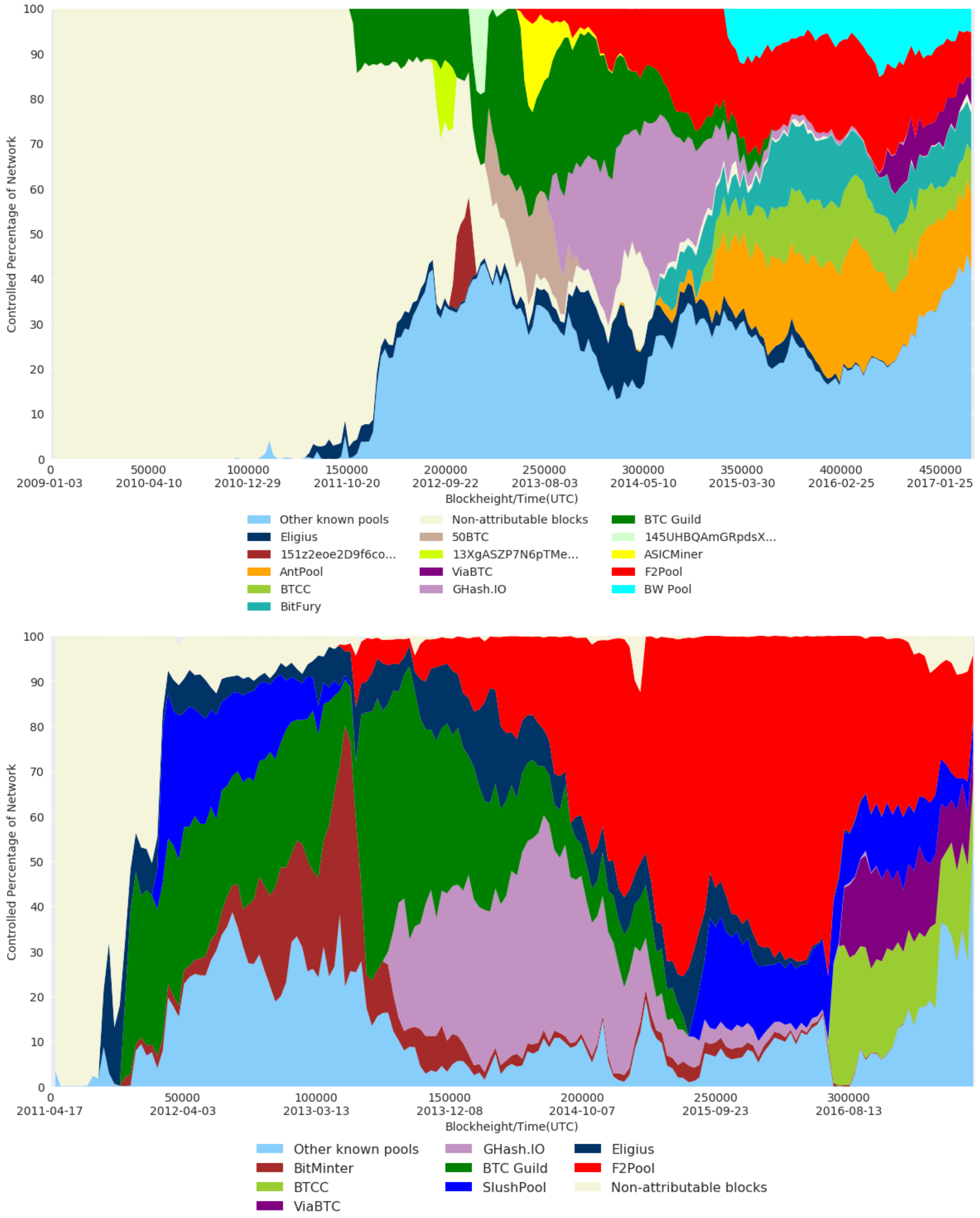


Figure 4.12: Distribution of blocks in Bitcoin (above) per pool over time compared to Namecoin (below) since launch of the respective cryptocurrency. Each data point resembles the share among 2,016 blocks (∼2 weeks), i.e., the difficulty adjustment period.

CHAPTER 5

Discussion

In this chapter we provide discussion on the implications of our findings. Section 5.1 focuses on the evaluation of the security implications related to merged mining, while an overview of possible solution approaches is provided in Section 5.2. Furthermore, we critically reflect the approach of our study and discuss potential shortcoming and possible improvements thereof in Section 5.3.

5.1 Security Implications

In this section we discuss the security implications of merged mining on the ecosystem of cryptocurrencies and study how current theoretic arguments, as described in [59], relate to our findings.

5.1.1 Introduction of New Attack Vectors

The advantage of merged mining is that miners are no longer forced to choose between mining one cryptocurrency or another. However, its biggest strength can also be viewed as a potential attack vector [59]. The ability to generate blocks for the merge-mined child blockchains at almost no additional cost, apart from maintaining a client node, allows misbehaving miners to carry out attacks without risking financial losses in both the parent and other child blockchains. Such an attack was carried out by the Eligius mining pool in 2012. Without their explicit consent, its miners were coerced to participate in an attack led by the pool operator, ultimately stalling the operation of the fledgling cryptocurrency CoiledCoin by mining empty blocks[1]. This attack serves as the predominant example for highlighting threats posed by merged mining on child cryptocurrencies: the miners of the pool did not suffer any financial loss and, as it appears, were not even aware of the attack, as all actions were performed solely by the operator.

[1] cf. `https://bitcointalk.org/index.php?topic=56675.msg678006#msg678006`

5.1.2 Centralization risks

Merged mining does not increase the costs to the miner with regards to solving the proof-of-work puzzle, which is considered to be the primary cost factor in PoW cryptocurrencies. However additional costs regarding bandwidth, storage and validation of the merge-mined blockchain's blocks/transactions are incurred regardless of the relative size or hash rate of the miner. Therefore, according to [59], merge-mined cryptocurrencies have a greater risk of centralization or concentration of mining power (economies of scale).

Our analysis indicates that merge-mined child blockchains experienced prolonged periods where individual mining pools have held shares beyond the theoretical bounds that guarantee the security of the cryptocurrency. We conclude that *current merge-mined currencies have a trend towards centralization.* However, it is too early to tell if the centralization trend also applies to multi-merged-mining in cryptocurrencies such as Myriadcoin. Multi-merge-mined blockchains allow for more than one parent cryptocurrency and have a greater chance to acquire a higher difficulty per PoW algorithm, in comparison to the respective parent blockchain. This, in fact, may change the underlying (crypto)economic assumptions with regards to merged mining and introduces new directions for research in this field.

The theoretic implications of a dishonest miner holding a large share of the network hash rate are well known [7,28,40,60]. However, we are not aware of any recent case where such an attack has been carried out in one of the analysed cryptocurrencies, as such evidence cannot easily be derived solely by analyzing the blockchain data structures. Rather, active measurements within the P2P network of the cryptocurrency are necessary [40]. Our analysis serves as a cautionary note – the impact of such an attack on the cryptocurrency market and the mining ecosystem are unclear. The apparent lack of cryptographically verifiable attribution information regarding the hash rate of mining pools only renders the situation worse. This bares additional risks of intended or unintended misattribution of non negligible fractions of the overall hash rate.

Furthermore, we want to point out that through the alternative use-cases of some of the merge-mined cryptocurrencies, certain attacks may also have additional implications. Namecoin for example, can be used to register and update arbitrary name-value pairs, such as DNS entries. In this case, every registered domain expires after a certain number of blocks (i.e., amount of time). Should a mining pool hold a large block share at that time, it can take over a domain name by blocking the required update (refresh) transaction to enter the blockchain in time. Once the domain name has expired, the misbehaving pool can register the domain himself.

5.1.3 Validation disincentive

Not only the detection of misbehaving pools with large hash rates requires active network monitoring, but also the verification of the *validation disincentive* assumption: In [59] the authors propose that miners which participate in merged mining have an incentive to skimp on (transaction) validation, since it becomes the main (computational) cost driver in merged mining. Although not mentioned explicitly in [59], the rate of blockchain forks, i.e., stale block rate of merged mined cryptocurrencies, could be an indicator for relaxed transaction validation of miners. Since stale blocks are not directly recorded in the blockchain, the only way to acquire the required measurements is through active monitoring of the involved peer-to-peer networks, as demonstrated in [18,19]. Conducting these measurements for multiple merge-mined cryptocurrencies is topic for future work. In addition, it might be necessary to actively trigger those conditions by broadcasting incorrect transactions/blocks. However, we stress that performing such tests in live networks raises ethical and financial questions.

5.1.4 Long-term dependency

Merged mining was originally conceived as a bootstrapping technique for alternative cryptocurrencies [59,72]. To the best of our knowledge, once introduced, no cryptocurrency has abandoned merged mining – not even the child cryptocurrencies which our analysis in Section 4.4 has shown to suffer from centralization issues. This observation indicates that using merged mining to bootstrap a new cryptocurrency and consequently switch to a different PoW algorithm or mandate solo mining, once a large enough community of active miners has formed, is not a viable option in practice. Hence, we argue that although merged mining can increase the hash rate of child blockchains, *it is not conclusively successful as a bootstrapping technique.*

5.2 Mitigating Centralization Issues

Merged mining was introduced as a mechanism to prevent fragmentation of computational resources and to help boostraping new cryptocurrencies. However, in this thesis we have shown that, while merged mining may be capable of achieving these goals, it leads to mining power centralization issues in implementing cryptocurrencies, potentially compromising the security of the underlying systems. In this section we discuss possible improvements to the concept of merged mining, helping to mitigate the related centralization problems.

5.2.1 Supporting N Proof-of-Work Algorithms

Huntercoin and Myriadcoin represent the first two cryptocurrencies supporting merged mining with multiple parent cryptocurrencies. However, in the case of Huntercoin we have seen that there exist mining pools capable of accumulating significant mining power shares across multiple cryptocurrencies with different proof-of-work algorithms. Myriadcoin extends the idea of Huntercoin by implementing five distinct proof-of-work algorithms, allowing merged mining with SHA256 and Scrypt, and adding additional guards to the consensus mechanism. As such, nodes compliant to the Myriadcoin rules will accept no more than five consecutive blocks generated through mining of the same PoW algorithm.

Still, a miner or mining pool dominating mining on two or more PoW algorithms will be capable of circumventing these rules, hence potentially being able to perform attacks such as double spending. Hence it remains yet unclear whether similar centralization effects as occurring in Huntercoin will be observable in Myriacoin and other cryptocurrencies relying on this approach in the future. Furthermore, apart from potential difficulties to scaling up to N parallel PoW algorithms, each new component potentially taken from exterior sources can compromise the security of the implementing blockchain. Once deployed, making successful changes to such consensus critical elements is likely to represent a difficulty task. As backward compatibility is not possible when modifying the consensus mechanism, the support of the majority of the participating nodes is necessary to prevent a permanent bifurcation of the network.

It is, however, important to mention that multi-PoW blockchains are expected to be able to better handle scenarios where a PoW algorithm fails to provide sufficient security guarantees: It is arguably more feasible to replace one PoW algorithm out of N, i.e., while N-1 other PoW algorithms remain active, compared to replacing the PoW algorithm of a single-PoW system. In fact, Myriadcoin has successfully been able to replace one of it's PoW algorithms, which became dominated by ASIC mining controlled by a small set of entities, aiming to prevent mining power centralization in 2016.

5.2.2 Relying on Less Adopted Proof-of-Work Algorithms

A potential approach to mitigate the observed centralization issues is to rely on less popular proof-of-work algorithms, which in contrast to SHA256 and Scrypt are not yet known to be dominated by large mining pools. A such approach was taken by Unitus [77], an recently launched cryptocurrency which was created by forking the code of Myriadcoin [2]. Unitus is a multi-PoW and multi-merge-mined cryptocurrency, supporting five different PoW algorithms, all of which can be used for merged mining. Reasoning by pointing towards the problems seen in other multi-merge-mined cryptocurrencies, Untius abstained from supporting SHA256 and Scrypt as PoW algorithms.

However, while a such approach may make it difficulty and infeasible for a single entity to accumulate and control the majority of mining power, it provides no strong theoretical guarantees. The rate of adoption of PoW algorithms by miners is linked to the popularity and economic value of the implementing cryptocurrencies. Should there be enough economic incentive for miners and mining pools to extend their operations to new algorithms, the latter will potentially face similar developments as SHA256 and Scrypt. Furthermore, due to potential lack of testing and review by the community, less adopted proof-of-work algorithms may contain shortcomings in terms of security assumptions and implementation detail, increasing the risk of compromising the underlying system.

[2] Due to the active development process at the time of writing, we did not include Unitus in the study performed in this thesis. However, an analysis of this cryptocurrency is planned as future work.

5.2.3 Verification Nodes and SPV Merged Mining

A different approach to mitigate centralization issues is to reduce the costs of merged mining and hence potentially incentivize higher participation of miners. Currently, miners must maintain a full node in the network of each merge-mined cryptocurrency, validating unconfirmed transactions and potentially storing the complete history of the blockchain. This not only increases the costs for participating miners but also may have negative effects on the health of the system, as it relies on the honest behaviour of miners which may or may not have a stake in the underlying cryptocurrency.

Hence, a possible approach would be to construct a model which allows SPV (Simplified Payment Verification) merged mining, i.e., separate the validation of transactions and maintenance of a consistent state of the blockchain from the process of merged mining. In this sense, we propose to introduce *validator* nodes responsible to collecting and validating the set of transactions to be included in the next block. Miners, deciding to merge-mine the cryptocurrency, are only required to run a SPV node in its network, which does not perform any validation tasks. Instead, miners connect to validation nodes and request the *template* for the next block, which already contains a set of validated transactions. As a result, miners are only required to include the block hash of the provided block template in the block header of the parent chain and follow the merged mining process as described in Section 2.2.1. A simplified visualization of a possible set up is provided in Figure 5.1.

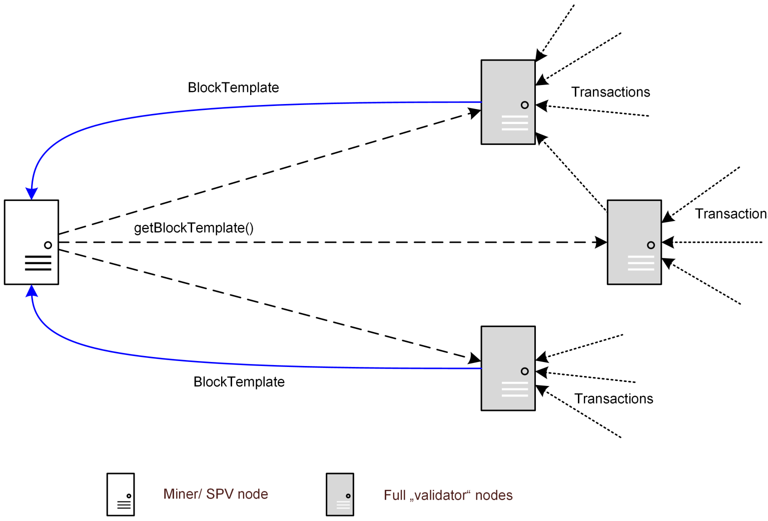


Figure 5.1: Simplified visualization of a SPV merged mining set up.

To provide incentive for honest behaviour, validator nodes receive a portion of the block reward and transaction fees. A possible way of guaranteeing payout is for validators to include their own address as output of the coinbase transaction, alongside that of the requesting miner. The validator then forwards the preliminary block hash, the coinbase transaction, as well as the Merkle tree root and the Merkle tree branch, necessary to verify that the coinbase transaction is part of the block, to the miner. While the miner can now verify the distribution of the block reward and transactions fees in the block, he cannot make any changes, without constructing a new block himself.

We note that the naive implementation described here contains numerous unresolved questions. As such, it must be specified how miners can and should decide which block template to use, which validators to trust and how to resolve conflicts, since it is highly probable different validator nodes will have different views on the state of the blockchain. In addition, relying on the honest behaviour of a small set of validators may yet again lead to centralization risks in the system. Furthermore, a fair reward distribution scheme must be derived, capable of maintaining incentives for honest behaviour of both miners and validators. Since a detailed study of the incentive models and potential new attack vectors, necessary to fully determine to viability of this concept, goes beyond the scope of this thesis, we dedicate the evaluation of this approach to future work.

5.2.4 Increase Maturity Phase for Rewards of Merge-Mined Blocks

In Bitcoin, newly generated coins cannot be transferred to other addresses during the so called *coinbase maturity phase* of 100 blocks (approx. 16.7h) [15]. The main purpose behind this rule is to prevent conflicts and inconsistencies in case a major bifurcation of the blockchain occurs. Similar, this "cool down" phase has been implemented in the other cryptocurrencies studied in this thesis, as summarized in Table 5.1.

However, this mechanism can also be used to prevent miners and mining pools from instantly selling the earned units of the cryptocurrency. During the coinbase maturity period, miners have a stake in the health of the mined blockchain, as events such as security issues may impact the economic value of the underlying currency. Hence, if the maturity period of merge-mined blocks were to be significantly increased, large mining pools would have less incentive to undertake malicious actions while participating in the mining process.

We note this approach does not solve the problem of mining power centralization but instead aims to further incentivize honest behaviour, especially of large miners and pools, capable of successfully performing attacks. However, this measure can be circumvented by, for example, directly exchanging keys or potentially locking funds (c.f. atomic hash locks [8, 16]). Furthermore, this approach may appear ineffective for cryprocrrencies with low economic value or against miners and mining pools having non-financial interests in malicious actions.

Table 5.1: Coinbase maturity phases in the studied cryptocurrencies.

Blockchain	Blocks	Block interval (min)	Time (approx.)
Bitcoin	100	10	16.7h
Namecoin	100	10	16.7h
Litecoin	100	2.5	4.2h
Dogecoin	240	1	4h
Huntercoin	100	1	1.6h
Myriadcoin	100	1	1.6h

5.3 Critical Reflection of the Performed Study

In this Section we provide a critical reflection of the study performed in this thesis, discussing performance and scalability of the blockchain extraction tool in Section 5.3.1 and challenges faced by our block attribution scheme in Section 5.3.2.

5.3.1 Performance and Scalability of the Blockchain Extraction Tool

Thorough empirical analysis of cryptocurrencies in most cases requires large scale collection of data, thereby parsing the underlying blockchain and mapping the data to a processable scheme. In the course of our study we extracted data from six different cryptocurrencies, thereby making use of the APIs of the reference client implementations. The *blockchain data miner* tool (c.f. Section 3.2), implemented tool to accomplish this task, was thereby built to support data collection in Bitcoin-like cryptocurrencies, i.e., systems using the same or similar codebase. We note, however, that minor changes to the code are still necessary when adding support for a new cryptocurrency. Furthermore, due to the active development process observed in such systems, breaking changes are not uncommon and have been experienced numerous times throughout our study. As such, monitoring the release cycles of cryptocurrencies is a task of great importance to ensure correct functionality of the extraction tool. Unfortunately, while widely adopted systems like Bitcoin, maintain a viable community, strict review processes and structured release cycles, less adopted cryptocurrencies often are unable to meet these standards. In fact, one-man development teams are not uncommon in this field.

A drawback identified in the process of evaluating the collected data is the performance of the utilized relational database format. While relational schemes allow for potent querying and complex joining of datasets, significant performance losses have been observed when working with large data sets. In addition, the required storage space by far exceeds that of the format currently used by Bitcoin and most cryprocurrencies. For example, storing the full Bitcoin blockchain at the time of writing requires approx. 150 GB of disk space. When mapped to a relational scheme, the necessary storage space increases to approx. 500 GB. The case is similar for Namecoin: while the blockchain size amounts to approx. 4.5 GB, the required disk space sums up to 17 GB when persisted in a relational database. Hence, optimizations and pruning of collected data was necessary to ensure acceptable performance in the course of this study. As such, we opted to only persist the coinbase transaction of each block, as the complete transaction history was not necessary for our analysis. The sizes of the resulting databases are summarized in Table 5.2.

Table 5.2: Summary of database sizes and extracted records for the cryptocurrencies evaluated in this study. The number of transactions approximately coincides with the number of extracted blocks, as only the coinbase transactions were persisted.

Blockchain	**Blocks**	**Inputs**	**Outputs**	**Addresses**	**Database size**	**Blockchain size**
Bitcoin	471,893	472,365	2,021,284	271,665	1577 MB	150 GB
Namecoin	347,176	347,524	347,561	187,571	2325 MB	4.5 GB
Litecoin	1,224,534	1,225,758	4,493,440	312,598	3683 MB	8.1 GB
Dogecoin	1,763,525	1,765,288	3,901,256	37,413	7116 MB	23 GB
Huntercoin	1,788,999	1790789	1790789	616386	18 GB	21 GB
Myriadcoin	2,089,975	2,092,065	4,651,417	297,681	8338 MB	4.2 GB

Potential performance improvements of the extraction tool can be achieved by directly parsing the .dat files used to store blockchain data by Bitcoin and the other evaluated cryptocurrencies. However, due to lack of documentation, this may prove to be a time consuming approach, as there is no guarantee the data format is similar for all of the observed systems. Furthermore, future changes to the storage mechanisms of the cryptocurrencies may yield a tool relying on specific formats unusable. The performance in the context of data evaluation can be improved by utilizing graph databases, e.g., in cases where the relations between different payout addresses are of interest.

5.3.2 Challenges faced by the Block Attribution Scheme

We acknowledge two possible challenges for our attribution scheme that might hamper its accuracy. First, the markers do not contain a cryptographic proof of identity. Hence, multiple miners might use the same marker or a miner might have mistyped or changed a marker at some point. While we did not observe any of the above cases in our dataset during semi-automated checks, there still remains a slight probability that a marker is actually a variant of another one.

Second, a miner might use a coinbase transaction to immediately split the block rewards to multiple outputs. As a consequence, we cannot attribute the original miner, unless they have used an already-known marker as well. We further cannot attribute blocks which contain unknown reward payout addresses, i.e., addresses that appear in the coinbase transaction of only this block. In these cases, we classify the block as *non-attributable.*

To the best of our knowledge, when relying on publicly-available blockchain data, there exists no technique to identify all the miners with 100% accuracy, if miners decide to not disclose their identity. Hence, some percentage of non-attributable blocks is to be expected.

Improvements can be achieved by setting up a network of nodes capable of performing live monitoring of the underlying blockchain. By collecting information on forks and unconfirmed transactions, it may be possible to detect significant behavioural patterns of large mining pools as well as attack attempts in general. However, as the geo-location of nodes may have impact of the collected data, a global set up would be required to achieve consistent and reliable metrics. As a such approach would require significant investments in infrastructure and would go beyond the scope of this study, we identify this as subject for future work.

A further improvement to our scheme can be made by tracing the flow of newly generated cryptocurrency units. While in our study we only use the reward payout address included in the coinbase transaction, following the money across multiple transactions may disclose even more address clusters attributable to distinct entities, possibly shedding light on relations between miners and mining pools, which until now appeared unrelated. First attempts to undertake evaluations of the money flow in Bitcoin are described in [49]. However, at the moment of writing we are not aware of successful money flow tracking frameworks capable of linking addresses across multiple cryptocurrencies.

CHAPTER 6

Conclusion and future work

In this thesis, we provided an overview of proof-of-work reusing mechanisms, including a detailed technical description of merged mining. Consequently, we tested current theories regarding merged mining from an empirical point of view and contributed to the discussion by raising new questions and directions for future work.

We derived an attribution scheme capable of linking blocks to the original miners and attribute mining activities across multiple ledgers. As a result, we achieved to map a significant portion of the mining pool ecosystem of the analyzed cryptocurrencies, beyond what was publicly known until now. The collected information sheds light on the long-term evolution of merged mining in different cryptocurrencies. While merged mining is a common practice in the cryptocurrency space, the empirical evidence suggests that only a small number of mining pools is involved in merged mining. These pools enjoy block shares beyond the desired security and decentralization goals.

In addition, we discussed the security implications of the observed issues related to merged mining and outlined possible mitigation strategies. However, It remains unclear and topic of future research whether new constructs, such as multi-merged mining, will succeed in resolving the outlined issues.

The multi-purpose usage of PoW in merged mining is an interesting application, not only from a resource consumption point-of-view, but also in the context of future data sharding and scalability discussions. Therefore, further research and analysis regarding merged mining is required as a basis for developing and building solutions, which will be able to stand the test of time.

List of Figures

List of Tables

Bibliography

[1] Coinmarketcap. http://coinmarketcap.com/. Accessed 2017-09-28.

[2] P2pool. http://p2pool.org/. Accessed: 2017-05-10.

[3] M. Ali, J. Nelson, R. Shea, and M. J. Freedman. Blockstack: Design and implementation of a global naming system with blockchains. http://www.the-blockchain.com/docs/BlockstackDesignandImplementationofaGlobalNamingSystem.pdf, 2016. Accessed: 2016-03-29.

[4] G. Andersen. Comment in "faster blocks vs bigger blocks". https://bitcointalk.org/index.php?topic=673415.msg7658481#msg7658481, 2014. Accessed: 2017-05-10.

[5] G. Andersen. [bitcoin-dev] weak block thoughts... https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-September/011157.html, 2015. Accessed: 2017-05-10.

[6] L. Anderson, R. Holz, A. Ponomarev, P. Rimba, and I. Weber. New kids on the block: an analysis of modern blockchains. http://arxiv.org/pdf/1606.06530.pdf, 2016. Accessed: 2016-07-04.

[7] E. Androulaki, S. Capkun, and G. O. Karame. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. In *CCS*, 2012.

[8] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille. Enabling blockchain innovations with pegged sidechains. http://newspaper23.com/ripped/2014/11/http-_____-___-_www___-blockstream___-com__-_sidechains.pdf, 2014. Accessed: 2017-09-28.

[9] A. Back et al. Hashcash - a denial of service counter-measure. http://www.hashcash.org/papers/hashcash.pdf, 2002. Accessed: 2017-09-28.

[10] S. Barber, X. Boyen, E. Shi, and E. Uzun. Bitter to better - how to make bitcoin a better currency. In *Financial cryptography and data security*, pages 399–414. Springer, 2012.

[11] J. Becker, D. Breuker, T. Heide, J. Holler, H. P. Rauer, and R. Böhme. Can we afford integrity by proof-of-work? scenarios inspired by the bitcoin currency. In *WEIS*. Springer, 2012.

[12] I. Bentov, R. Pass, and E. Shi. Snow white: Provably secure proofs of stake. `https://eprint.iacr.org/2016/919.pdf`, 2016. Accessed: 2017-09-28.

[13] Bitcoin Community. Bitcoin developer guide- transaction data. `https://bitcoin.org/en/developer-guide#term-merkle-tree`. Accessed: 2017-06-05.

[14] Bitcoin Community. Bitcoin protocol documentation - merkle trees. `https://en.bitcoin.it/wiki/Protocol_documentation#Merkle_Trees`. Accessed: 2017-06-05.

[15] Bitcoin community. Bitcoin protocol rules. `https://en.bitcoin.it/wiki/Protocol_rules`. Accessed: 2017-08-22.

[16] V. Buterin. Chain interoperability. Technical report, Tech. rep. 1. R3CEV, 2016.

[17] W. Dai. bmoney. `http://www.weidai.com/bmoney.txt`, 1998. Accessed: 2017-09-28.

[18] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013.

[19] C. Decker and R. Wattenhofer. Bitcoin transaction malleability and mtgox. In *Computer Security-ESORICS 2014*, pages 313–326. Springer, 2014.

[20] Dogecoin community. Dogecoin reference implementation. `https://github.com/dogecoin/dogecoin`. Accessed: 2017-09-28.

[21] J. R. Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[22] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*, pages 139–147. Springer, 1992.

[23] I. Eyal. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 89–103. IEEE, 2015.

[24] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[25] H. Finney. Reusable proofs of work (rpow). `http://web.archive.org/web/20071222072154/http://rpow.net/`, 2004. Accessed: 2017-09-28.

[26] P. Franco. *Understanding Bitcoin: Cryptography, engineering and economics.* John Wiley & Sons, 2014.

[27] A. Gervais, G. Karame, S. Capkun, and V. Capkun. Is bitcoin a decentralized currency? volume 12, pages 54–60, 2014.

[28] 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. ACM, 2016.

[29] I. Giechaskiel, C. Cremers, and K. B. Rasmussen. On bitcoin security in the presence of broken cryptographic primitives. In *European Symposium on Research in Computer Security (ESORICS)*, September 2016.

[30] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Performance Evaluation*, 104:23–41, 2016.

[31] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 129–144, 2015.

[32] Huntercoin developers. Huntercoin reference implementation. `https://github.com/chronokings/huntercoin`. Accessed: 2017-06-05.

[33] B. Jakobsson and A. Juels. Proofs of work and bread pudding protocols, Apr. 8 2008. US Patent 7,356,696; Accessed: 2017-06-05.

[34] M. Jakobsson and A. Juels. Proofs of work and bread pudding protocols. In *Secure Information Networks*, pages 258–272. Springer, 1999.

[35] A. Judmayer, N. Stifter, K. Krombholz, and E. Weippl. Blocks and chains: Introduction to bitcoin, cryptocurrencies, and their consensus mechanisms. *Synthesis Lectures on Information Security, Privacy, & Trust*, 9(1):1–123, 2017.

[36] A. Juels and J. G. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *NDSS*, volume 99, pages 151–165, 1999.

[37] A. Juels and B. S. Kaliski Jr. Pors: Proofs of retrievability for large files. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 584–597. Acm, 2007.

[38] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan. An empirical study of namecoin and lessons for decentralized namespace design. In *WEIS*, 2015.

[39] G. O. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 906–917. ACM, 2012.

[40] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun. Misbehavior in bitcoin: A study of double-spending and accountability. volume 18, page 2. ACM, 2015.

[41] A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[42] S. King. Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th*, 2013.

[43] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90, 2016.

[44] Lerner, Sergio D. Rootstock plattform. `http://www.the-blockchain.com/docs/Rootstock-WhitePaper-Overview.pdf`. Accessed: 2017-06-05.

[45] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 919–927. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[46] Litecoin community. Litecoin reference implementation. `https://github.com/litecoin-project/litecoin`. Accessed: 2017-09-28.

[47] I. Maven. Apache maven project, 2011.

[48] G. Maxwell. Comment in "[bitcoin-dev] weak block thoughts...". `https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-September/011198.html`, 2016. Accessed: 2017-05-10.

[49] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140. ACM, 2013.

[50] S. Micali. Algorand: The efficient and democratic ledger. http://arxiv.org/abs/1607.01341, 2016. Accessed: 2017-02-09.

[51] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz. Permacoin: Repurposing bitcoin work for data preservation. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 475–490. IEEE, 2014.

[52] A. Miller, A. Kosba, J. Katz, and E. Shi. Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 680–691. ACM, 2015.

[53] B. Momjian. *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley New York, 2001.

[54] Myriad core developers. Myriadcoin reference implementation. `https://github.com/myriadcoin/myriadcoin`. Accessed: 2017-06-05.

[55] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`, Dec 2008. Accessed: 2017-09-28.

[56] S. Nakamoto. Merged mining specification. `https://en.bitcoin.it/wiki/Merged_mining_specification`, Apr 2011. Accessed: 2017-09-28.

[57] Namecoin Community. Merged mining. `https://github.com/namecoin/wiki/blob/master/Merged-Mining.mediawiki#Goal_of_this_namecoin_change`. Accessed: 2017-08-20.

[58] Namecoin community. Namecoin reference implementation. `https://github.com/namecoin/namecoin`. Accessed: 2017-09-28.

[59] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton University Press, 2016.

[60] K. Nayak, S. Kumar, A. Miller, and E. Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *1st IEEE European Symposium on Security and Privacy, 2016.* IEEE, 2016.

[61] K. J. O'Dwyer and D. Malone. Bitcoin mining and its energy footprint. 2014.

[62] 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.

[63] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[64] Pseudonymous("TierNolan"). Decoupling transactions and pow. `https://bitcointalk.org/index.php?topic=179598.0`, 2013. Accessed: 2017-05-10.

[65] P. R. Rizun. Subchains: A technique to scale bitcoin and improve the user experience. *Ledger*, 1:38–52, 2016.

[66] K. Rosenbaum. Weak blocks - the good and the bad. `http://popeller.io/index.php/2016/01/19/weak-blocks-the-good-and-the-bad/`, 2016. Accessed: 2017-05-10.

[67] K. Rosenbaum and R. Russell. Iblt and weak block propagation performance. *Scaling Bitcoin Hong Kong (6 December 2015)*, 2015.

[68] M. Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.

[69] M. Rosenfeld. Analysis of hashrate-based double spending. `http://arxiv.org/abs/1402.2009`, 2014. Accessed: 2016-03-09.

[70] R. Russel. Weak block simulator for bitcoin. `https://github.com/rustyrussell/weak-blocks`, 2014. Accessed: 2017-05-10.

[71] A. Sapirshtein, Y. Sompolinsky, and A. Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.

[72] Sathoshi Nakamoto. Comment in "bitdns and generalizing bitcoin" bitcointalk thread. `https://bitcointalk.org/index.php?topic=1790.msg28696#msg28696`. Accessed: 2017-06-05.

[73] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. In *FC '16: Proceedings of the the 20th International Conference on Financial Cryptography*, February 2016.

[74] B. Sengupta, S. Bag, S. Ruj, and K. Sakurai. Retricoin: Bitcoin based on compact proofs of retrievability. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, page 14. ACM, 2016.

[75] N. Szabo. Bit gold. `http://unenumerated.blogspot.co.at/2005/12/bit-gold.html`, 2005. Accessed: 2017-09-28.

[76] M. B. Taylor. Bitcoin and the age of bespoke silicon. In *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, page 16. IEEE Press, 2013.

[77] Unitus developers. Unitus reference implementation. `https://github.com/unitusdev/unitus`. Accessed: 2017-08-22.

[78] M. Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security*, pages 112–125. Springer, 2015.

[79] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis, and S. Deleuze. Spring boot reference guide. Technical report, 2013-2016.

[80] A. Zamyatin. Name-squatting in namecoin. (unpublished BSc thesis, Vienna University of Technology), 2015.

APPENDIX A

Appendix

A.1 Development of Mining Power Shares in Litecoin and Dogecoin

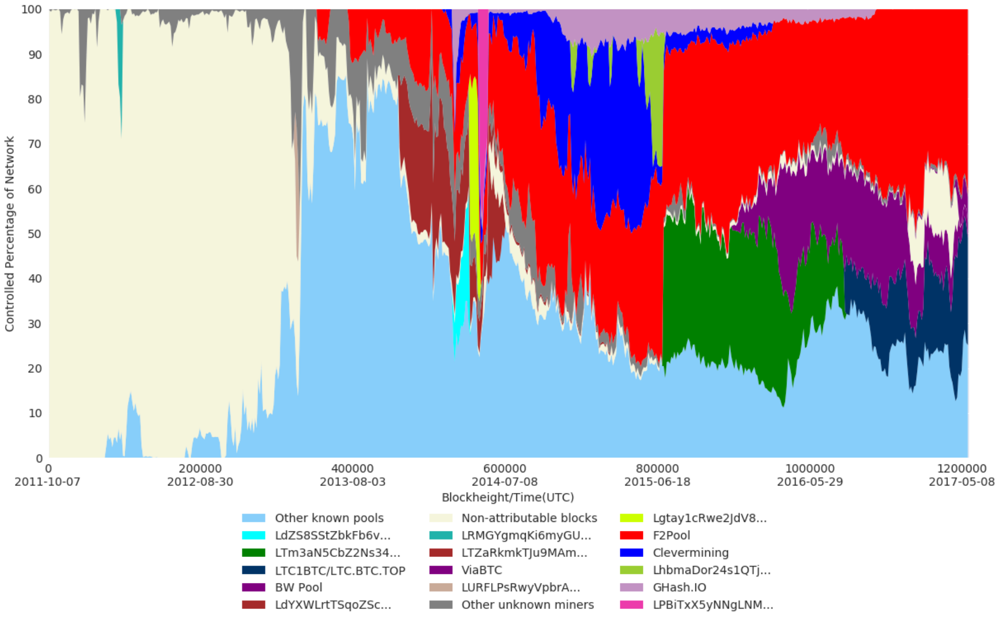


Figure A.1: Distribution of blocks in Litecoin per pool over time since launch. Each data point resembles the share among 1,440 blocks, i.e., the difficulty adjustment period in Litecoin (∼60h).

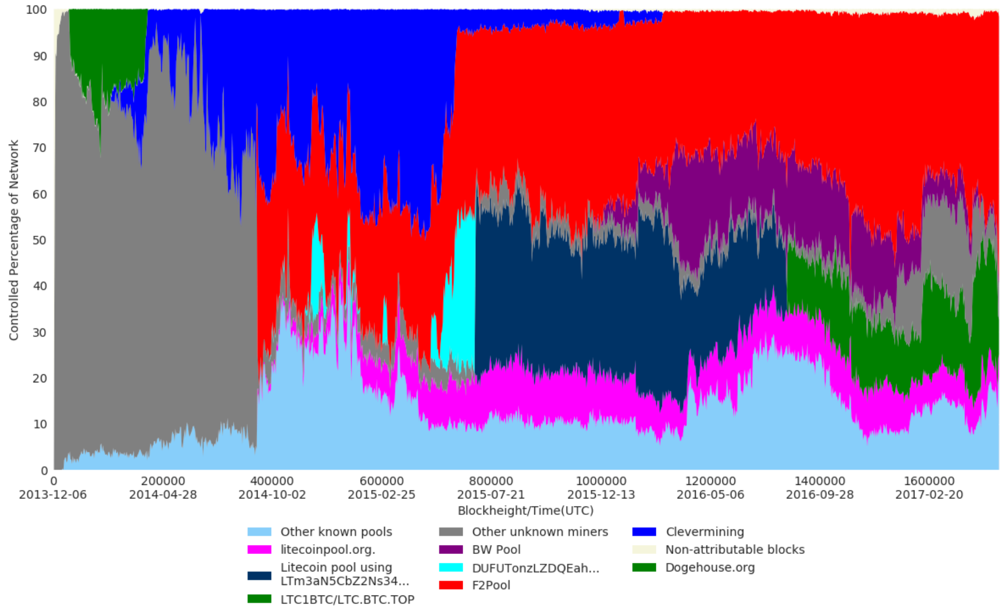


Figure A.2: Distribution of blocks in Dogecoin per pool over time since launch. Each data point resembles the share among 1,440 blocks, i.e., ∼24h. Note: in Dogecoin, the difficulty is recalculated after every block.

A.2 Development of Mining Power Shares in the Multi-PoW Cryptocurrencies Huntercoin and Myriadcoin

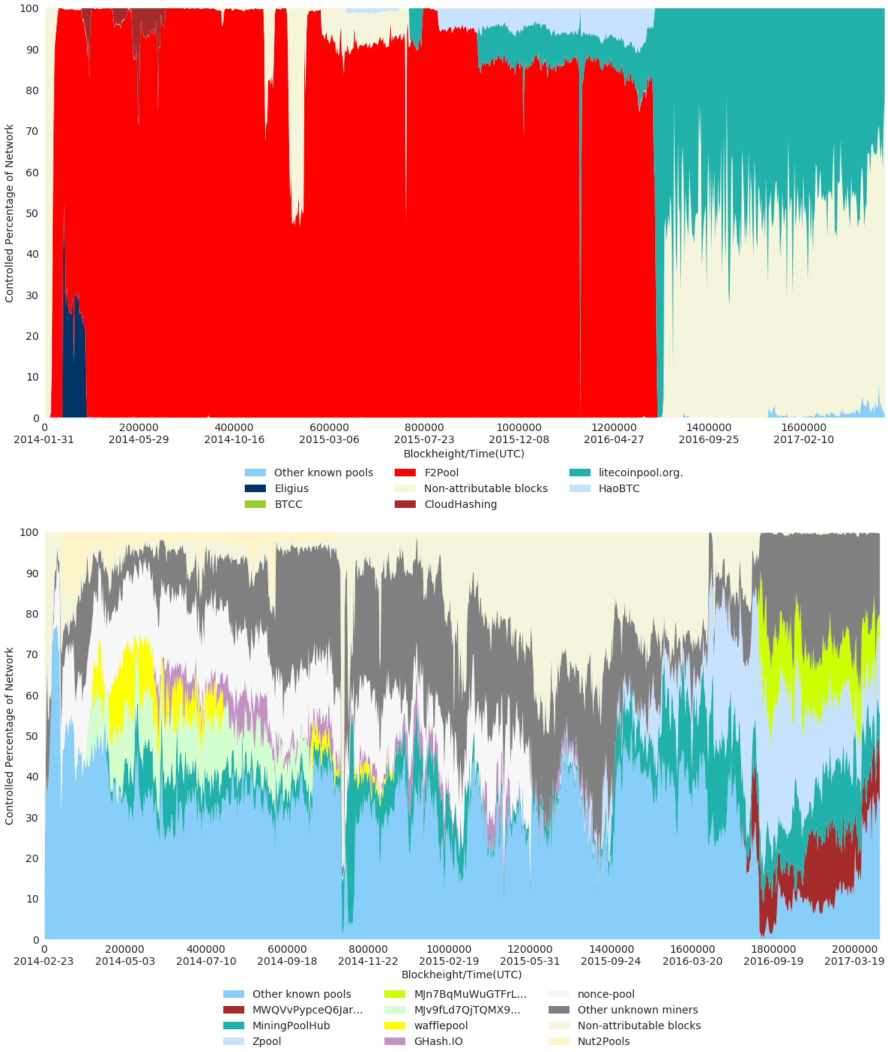


Figure A.3: Distribution of blocks in Huntercoin (above) per pool over time compared to Myriacoin (below) since launch of the respective cryptocurrency. Each data point resembles the share among 2,016 blocks, i.e., the difficulty adjustment period in Huntercoin (∼33h) and ∼33h in Myriadcoin. Note: in Myriadcoin, the difficulty is recalculated after every block.