Echoes of the Past: Recovering Blockchain Metrics From Merged Mining

Nicholas Stifter[1,3] ✉, Philipp Schindler[1], Aljosha Judmayer[1],
Alexei Zamyatin[2,1], Andreas Kern[1], Edgar Weippl[1,3]

[1]SBA Research, Vienna, Austria
[2]Imperial College London, United Kingdom
[3]Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, TU Wien
Email: (firstletterfirstname)(lastname)@sba-research.org

Abstract. So far, the topic of *merged mining* has mainly been considered in a security context, covering issues such as mining power centralization or cross-chain attack scenarios. In this work we show that key information for determining blockchain metrics such as the *fork rate* can be recovered through data extracted from merge mined cryptocurrencies. Specifically, we reconstruct a long-ranging view of forks and stale blocks in Bitcoin from its merge mined child chains, and compare our results to previous findings that were derived from live measurements. Thereby, we show that live monitoring alone is not sufficient to capture a large majority of these events, as we are able to identify a non-negligible portion of stale blocks that were previously unaccounted for. Their authenticity is ensured by cryptographic evidence regarding both, their position in the respective blockchain, as well as the Proof-of-Work difficulty.
Furthermore, by applying this new technique to Litecoin and its child cryptocurrencies, we are able to provide the first extensive view and lower bound on the stale block and fork rate in the Litecoin network. Finally, we outline that a recovery of other important metrics and blockchain characteristics through merged mining may also be possible.

1 Introduction

In blockchain-based cryptocurrencies the *fork rate* is considered to be an essential metric to better gauge the performance, capacity, and health of the respective communication network [1], and may also help in estimating other aspects such as their security [2] or degree of decentralization [3]. Furthermore, the fork rate can be indicative of adversarial behavior, such as *selfish mining* and its variants [2, 4–6] and other attacks that induce a higher ratio of stale blocks [7–9], or highlight periods of contention over protocol rule changes [10]. Historic and long-ranging data on stale blocks and the fork rate could also help determine the effectiveness of improvement measures and also provide a vital empirical basis for both predicting and directing future development.

However, for many cryptocurrencies such extensive data sets are not always readily available as a consequence of both design decisions, as well as the necessity to perform ongoing live monitoring to try and capture these events from gossip in the peer-to-peer (p2p) network. Moreover, while public sources of live monitoring data from popular

cryptocurrencies, such as Bitcoin, do provide information on stale blocks [11–13], it is not clear how extensive and well-connected these monitoring efforts were for the data to be considered representative. Finally, some of the available information may lack the necessary data to perform verification, such as establishing the validity of the respective Proof-of-Work.

In this paper we present a novel reconstruction technique for stale blocks that can be applied to Bitcoin-like Proof-of-Work blockchains, which have served the role as a *parent* chain for merged mining. Specifically, we shine light on the aspect that the prevalent implementation of merged mining requires the *child* blockchain to include both, the full block header, as well as the Merkle branch and coinbase transaction, of a candidate parent block *every time* a child block is produced through merged mining, to be able to validate its correctness.

Using Bitcoin as an example parent, we extract and analyze the additional data embedded through merged mining from several of Bitcoin's child currencies, and compare our findings to those of Decker and Wattenhofer [1] and other stale block and fork rate data derived from live measurements [11–13]. Based on this analysis, we are not only able to show that our technique is successful in recovering stale blocks and forks, but also that our method uncovers a non-negligible portion of blocks that have otherwise not been captured by live monitoring. This raises interesting new questions on the accuracy of former fork rate estimates and shows that the ratio of stale blocks is higher than previously anticipated. The contributions of this paper can be summarized as follows:

- We outline how the process of merged mining provides an interesting, but generally overlooked side channel for gaining additional information about the involved *parent* cryptocurrencies.
- We show that the data from merged mining can be used to recover stale blocks and forks in the parent chain, and may also enable the inference of other key blockchain metrics.
- Our analysis reveals a sizable portion of forks and stale blocks that were not recognized through live monitoring activities, suggesting that this new approach serves as a complementary mechanism for determining the fork rate. Furthermore, our findings suggest that previous models and estimates on fork rates and stale blocks should be re-evaluated.
- We demonstrate that our approach can be readily applied to other merge mined parent cryptocurrencies by reconstructing (a lower bound of) the fork- and orphan-block rate in Litecoin [14].

2 Background

First, this section outlines the concept and relevance of forks and stale blocks to Bitcoin-like cryptocurrencies, and why they can be considered key metrics, after which the core ideas and primitives related to merged mining are presented.

2.1 Forks and Stale Blocks

Simplified, in Bitcoin and similar cryptocurrencies, the *heaviest chain rule*, i.e. the chain with the most consecutive Proof-of-Work (PoW), determines which sequence of blocks is considered canonical and defines the ledger's current valid state [15]. In

this context, PoW puzzles that are based on blocks play a key role as part of the consensus mechanism through which Bitcoin achieves aspects of decentralization [16, 17]. Because the discovery of puzzle solutions, referred to as mining, is probabilistic, and also because of propagation delays in the underlying peer-to-peer network [1], it is possible that more than one block with a solution can exist for a particular height of the chain at the same time, leading to a so called *fork* in the blockchain. In this case miners may choose to extend either one of these valid chain tips. Assuming an honest majority of computational power[1] follows the heaviest chain rule, it can nevertheless be shown that eventual agreement (and other desirable properties) over a distributed ledger can be achieved as miners converge on a single common chain [17, 19].

Within this paper, we refer to any blocks which satisfy the prescribed PoW puzzle difficulty of the main chain at that time or height, but are not part of the canonical chain, as being *stale*. Further, we consider a *fork* to be a branch of stale blocks of length $n \geq 1$ that can be cryptographically linked to a block in the canonical main chain. On the other hand, blocks for which we cannot ascertain such a link are called *orphans*.

Blockchain metrics such as the *fork rate* and *ratio of stale blocks* can provide useful information about the health and current state of cryptocurrencies. A high stale block rate may be indicative of insufficient network or block validation capacities [1,20] and is detrimental to the overall security, as it can increase the likelihood of successful double-spending attacks [2]. Additionally, many described attacks, such as selfish mining and its variants, or attempts at double spending, have an impact on the stale block rate [2,5]. Finally, contentious or unsuccessful protocol changes may also manifest themselves in high stale block rates as a portion of the network may fall into disagreement and mine on different branches [10]. Here, it depends both on the protocol upgrade mechanism and rule set a node prescribes to if a block is only considered stale or deemed invalid and not considered at all.

However, the Bitcoin protocol and many of its direct derivatives do not provide mechanisms or incentives to include information on stale blocks and forks as part of the consensus layer, though it has been outlined that taking these aspects into consideration could improve protocol characteristics [15, 21] and would generally lead to underlying structures that form a directed acyclic graph instead of a chain [22, 23].

Furthermore, while protocol implementations do serialize information on stale blocks observed through p2p gossiping locally[2], due to privacy concerns that arise from the ability to fingerprint [25] a node based on the set of stale blocks it knows, a limit of thirty days is imposed on how far back stale blocks will be served to peers [24]. Hence, up until now, the primary source for both historic and current data related to stale blocks and forks for Bitcoin and similar cryptocurrencies comes from dedicated monitoring operations that gather and provide this additional information [1, 3, 11–13, 26].

2.2 Merged Mining

Merged mining is an approach, whereby miners can leverage on the same process for searching for a valid PoW solution in more than one cryptocurrency, without having to

[1] We exclude attacks such as *selfish mining* [4] and possible countermeasures [18] in this example to simplify the discussion.

[2] In Bitcoin core [24] the RPC command *getchaintips* can be used to list all forks and stale blocks the local node knows of.

split their computational resources among them. The motivation behind merged mining originally stemmed from the problem of how to avoid that competing cryptocurrencies reduce each other's security by competing in hash rate, and has also been suggested as a suitable bootstrapping and hardening mechanism for fledgling cryptocurrencies [27,28]. The idea of repurposing or reusing the computational effort spent in computing Proofs-of-Work is not new, and was first systematically described by Jakobson and Juels as *bread pudding protocols* [29].

The prevalent mechanism among existing cryptocurrencies by which merged mining is implemented follows a *parent* and *child* relationship. Thereby, no substantial changes to the block header and verification logic of already deployed cryptocurrencies is required. The hash of a candidate block in the child cryptocurrency is to be embedded into the candidate block of the parent in a prescribed way, generally within the coinbase transaction [30] of the block. Then the search for a valid PoW is performed on the parent's block header as usual. While such an approach necessitates the explicit support of merged mining in the child cryptocurrency, the parent can be oblivious to any ongoing merged mining activity, relating this protocol change to the concept of a *velvet fork* [10, 31].

This form of merged mining requires miners to additionally attach the block header and coinbase transaction (and its Merkle branch) of the parent to the block submitted to the child chain (see figure 1). These elements are necessary to validate the PoW performed on the header of the parent block, the so called Auxiliary PoW (*AuxPoW*). Thereby, merge mined blockchains contain additional information from their parents (see Fig. 3). The PoW difficulty for the child chain is usually lower than that of the parent chain [28] and is instead encoded and adjusted in the headers of the child chain blocks. Therefore, partial (also called weak or near) PoW solutions for a parent blockchain may nevertheless be valid for one or more child chains. If more than one child blockchain is to be merge mined with the same parent chain, a Merkle tree root hash as well as a parameter defining its size is included by the miner. The leaves of the tree represent the hashes of the block headers of each child blockchain. If merged mining involves only one child blockchain, the hash of the block header of the child blockchain can be included directly in the coinbase of the parent.

When mining multiple child chains, it is vital to ensure that merged mining does not occur for multiple forks of the same child blockchain; this would compromise the security of the latter as two branches of a fork can be mined at the same time. To address this issue, each child blockchain has a fixed `chainID` that is defined by its developers. For example, the `chainID` for Namecoin is set [32] to the value `0x0001`. Every miner can choose freely for how many and for which PoW child blockchains they want to perform merged mining and, hence, maintain a different Merkle tree. The combination of `MerkleSize`, `MerkleNonce`, and `chainID` are fed to a linear congruential generator so as to produce the unique position of a child blockchain `chainID` on a Merkle tree of a given size [33].

Merged mining was first introduced in Namecoin at block height 19200 (2011-10-11) and the corresponding AuxPoW built upon the Bitcoin block at height 148553. Since then merged mining has been deployed in a variety of other cryptocurrencies [28].

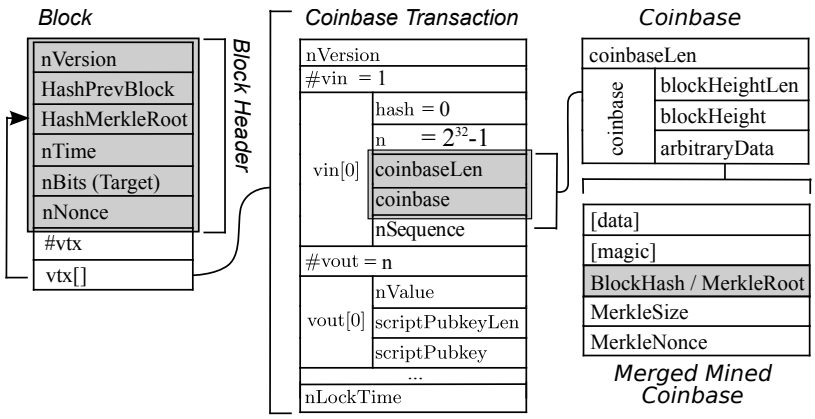

Fig. 1. Common Bitcoin block and merged mining data structures [30]

3 Merged Mining as a Side Channel

After having covered the principle mechanisms of merged mining, we first describe how the accrued data can serve as a side channel for gaining information about a parent chain, and then outline potential information that could leak this way.

3.1 Auxiliary Proofs-of-Work as Informants

The prevalent form of merged mining requires the child chain to include the block header, coinbase transaction and Merkle branch of the parent, otherwise it is not possible to verify the correctness of the Auxiliary Proof-of-Work. Hereby, the difficulty requirement for the child chain does not have to be equivalent to that of the parent, and other parametrizations, such as the target block interval, can also differ. For example, I0Coin [34], which can be merge mined with Bitcoin, has a target block interval of 90 seconds compared to the 600 second target of Bitcoin. This means that a valid AuxPoW for a child block may not necessarily be considered a valid PoW in the parent. In principle, the child cryptocurrency could even go as far as to change characteristics of the PoW itself while still retaining the ability for merged mining, such as reversing the final output bits of the utilized hash function, or additionally applying a bit mask before checking the output. For instance, Garay et al. [17] outlines how such *2-for-1 PoWs* can be achieved.

However, in practice the required PoW format is the same for child and parent, thereby encoding additional useful information regarding parent solution candidates because their discovery probability is no longer independent. An explanation for this behavior may be that the used mining hardware (ASICs) is not readily adaptable. If the difficulty requirement of the PoW for the parent, D_p, exceeds the difficulty for the child, D_c, i.e $D_p \geq D_c$, then finding a valid PoW in the parent will at the same render it a valid AuxPoW for the child. With few distinct exceptions, it is observed that merge mined child cryptocurrencies do not exceed the parent difficulty [28], and one would therefore expect a subset of valid PoWs that were mined in the parent to become encoded as AuxPoWs in the merge mined children.

Assuming merge miners are economically rational actors, it would be expected that the candidate block headers being mined in the parent cryptocurrency are intended to be valid, i.e., contain a valid previous block hash, time stamp and difficulty etc., as miners would otherwise be wasting computational resources without receiving compensation from successfully mining valid blocks. Because transactions are not embedded in the auxiliary PoW, its full validity can not be ascertained, unless it contained only the coinbase transactions. A miner does not, a priori, know when they will find a valid PoW solution and hence is incentivized to update and maintain a valid candidate block and its header while mining. As we later outline in the discussion of our findings in Section 6, we identified sporadic patterns in our data that are not easily explained under this assumption and may be indicative of software malfunction or misconfiguration.

3.2 Parent Block Information Leaking Potential

As previously outlined, the AuxPoW provides a snapshot of the particular miner's parent block header candidate at the time the child solution was found. Depending on both the block interval and difficulty requirement of the child chain, multiple such snapshots from different miners can exist between the discovery of blocks in the parent cryptocurrency, providing different vantage points of the network. Because the entire coinbase transaction is also available for each AuxPoW, miner identification schemes such as the approach from Judmayer et al. [28] are also applicable. Additionally, most valid merge mined PoWs of the parent chain are likely to be recorded in the child chain because the child block would also meet its respective difficulty requirement.

Further, in the case of a fork event in the parent chain, there is a chance that one, or even both, of the parent block headers are captured through AuxPoWs if they were merge mined, and consecutive stale blocks from a prolonged fork may also be recorded in child cryptocurrencies in this fashion. In this respect, being able to draw information from multiple merge mined children with different block intervals may increase the likelihood that the block headers of competing forks are present in at least one of them.

Another interesting aspect is the additional, and possibly better, timing information that can be gained through both the child block(s) directly linked to an AuxPoW, as well as the additional time stamps from candidate parent block headers.

Categorization of Recoverable Blocks: Based on the information available within an AuxPoW, we categorize recoverable parent block headers and illustrate their relationship to the canonical parent chain in terms of difficulty requirements in Figure 2.

- **Canonical Block:** If the block header belongs to a block that is part of the canonical main chain in the parent it is considered a canonical block.
- **Stale Block:** A block header that does not end up as part of the canonical parent chain but could have been a valid fork based on its (verifiable) difficulty and respective height or time stamp relative to the parent.
- **Near Block:** Parent block headers that do not meet the difficulty requirement of the canonical chain are referred to as near blocks. While near blocks are not valid in the parent chain, they may still provide useful information such as the particular miner's view of the longest chain at that time [3]

[3] Assuming the miner follows the protocol rule of extending the longest chain it knows of.

– **Orphan Block:** Blocks for which we are unable to establish a cryptographic link that eventually leads to a canonical block are considered orphan blocks. Orphan blocks have weaker guarantees as to their potential validity, as it is unclear if they were actually related to the parent chain being analyzed.
– **Shadow Block:** We refer to predecessors, where we can not obtain the full block header, e.g. only a hash, as shadow blocks. Even without the ability of cryptographic verification it can be possible to perform some basic validation, i.e, by checking if the hash itself could have met the required difficulty of the parent chain at the approximate time or height, which can be inferred from data in successors that build upon the shadow block.[4] Any parent headers that build upon a shadow block are implicitly *orphans* because they cannot be linked to the canonical chain.
– **Invalid Block:** Based on the information available in the AuxPoW, some parent block headers may be identified as invalid because they do not follow the prescribed protocol rules of the parent chain. For instance, the encoded target difficulty may be too low or the time stamp outside of the permissible range.

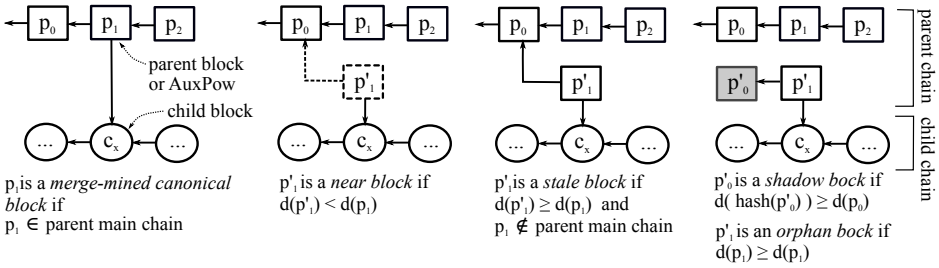

Fig. 2. Recoverable Block Categories and their Relationship to the Parent Chain.

4 Data Sources and Processing

In this work, we primarily consider *Bitcoin* [35], as it not only has the longest history of being a parent to merged mining, but also because there exists both live monitoring services that provide information on forks and stale blocks [11–13], as well as scientific literature that relates to forks and stale block rates [1,2,36,37]. Thereby, we gain access to necessary information for comparing and validating our results, for instance through forming the intersection of block headers that have been discovered by live monitoring and merged mining. Furthermore, we also apply our approach to *Litecoin* [14] to determine if it is readily adaptable to other merge mined cryptocurrencies.

Our raw blockchain data sources related to Bitcoin, Litecoin, and their merge mined children herein considered, are listed in Table 1 in the appendix, and were collected using fully validating clients. In total, we gathered data from 7 merge mined children for Bitcoin, and 2 merge mined children for Litecoin. Furthermore, we also included data

[4] In Litecoin and its children this validation is not possible because a DSHA256 hash of the block header is used for linking, instead of the scrypt hash used for the PoW.

from the Bitcoin Cash fork to help identify orphan blocks, because it has served as a parent for DSHA256 merge mined currencies. The set of merge mined cryptocurrencies we selected is not exhaustive, and the focus was placed on projects with a long history of merged mining in order to gain as extensive of a view as possible. Relevant blockchain and AuxPoW data was then extracted through the respective RPC interface of the cryptocurrency client and aggregated in a graph database (Neo4J [38]), to aid in our exploratory data analysis and simplify searching for interesting patterns.

To determine if the extracted AuxPoW block headers can be considered stale block candidates for the target parent chain, several steps were followed:

1. The encoded difficulty target in the AuxPoW header was checked against the resulting block hash to determine if the parent header forms a valid PoW[5].
2. To establish a time frame for Bitcoin difficulty epochs (2016 blocks), we consider the time stamp of the first block in the epoch as the starting point and the time stamp of the first block in the *next* epoch as its end.
3. A link between the AuxPoW and a particular difficulty epoch in the parent was established to determine if the PoW difficulty is high enough to be considered valid. This was first attempted based on the block height, which can either be inferred if the block is linked to the canonical chain or, if the block is BIP34 compliant [39], determined from the height encoded in the coinbase transaction.
4. If the height could not be inferred in the previous step, the time stamp in the block header is used instead. For shadow blocks, the lowest time stamp of any AuxPoW that builds on top of it was used.

In respect to the live monitoring data that was used to compare and evaluate our results against, we rely on different sources. First, we gathered publicly available data on forks and stale blocks from block explorers [11–13, 26]. Second, we reached out to the authors of academic measurement studies related to Bitcoin's fork rate and inquired if they could provide us with the relevant monitoring data, and were kindly provided data from [1]. See Table 2 in the appendix for more details on live monitoring data.

5 Analysis

To analyze the feasibility and effectiveness of merged mining as a side channel, we focus on the recovery of information related to a key metric in the parent, namely the *stale block rate* and hereby resulting *forks*. This is of particular value, as long-ranging views that estimate stale block and fork rates are not readily retrievable from the data persisted in the respective blockchain for most cryptocurrencies, and require additional live monitoring efforts.

We subsequently first compare our findings on stale blocks and forks in Bitcoin to the measurement study conducted by Decker and Wattenhofer [1], as the authors have kindly provided us with raw data that was used in their work. Following this initial evaluation, we then extend our analysis over a wider time span and draw upon multiple live monitoring data sources for comparison.

[5] We also validated if the AuxPoW actually meets the difficulty encoded in the child

5.1 Comparison to Decker and Wattenhofer Monitoring Data

The Bitcoin p2p measurement study of Decker and Wattenhofer (DW) has provided important insights on the blockchain fork rate and its dependency on propagation times, and serves as a critical reference point for Bitcoin's performance [20]. Therein, live monitoring data gathered over two 10000 block intervals, ranging from block height 180000 to 190000 and 200000 to 210000, was analyzed and compared to a formal model for predicting the probability of a blockchain fork. In particular, while the first monitoring interval only involved passive observation, the second interval was actively influenced by relaying block information to as many peers as possible. Thereby, it was empirically shown that propagation delay, and consequently also block size, plays an important factor in the probability of forks, as the fork rate dropped from a measured 1.69% in the first interval to 0.78% in the second interval. Furthermore, at a 1.78% predicted fork rate, the presented formal model using propagation metrics from the live measurements was relatively close to the actual monitored fork rate of the first interval.

Because the commencement of merged mining in Bitcoin dates back far enough to cover both intervals, an obvious approach would be to compare the fork rate recoverable through merged mining with these results. Unfortunately, while we were able to obtain a large portion of the raw monitoring captures from the respective authors, it was reported to us that some of the data was rendered unrecoverable due to storage failure. Specifically, we were unable to obtain any data related to the second monitoring interval. Nevertheless, a comparison of our recovered stale blocks with live monitoring from the first interval already reveals interesting insights.

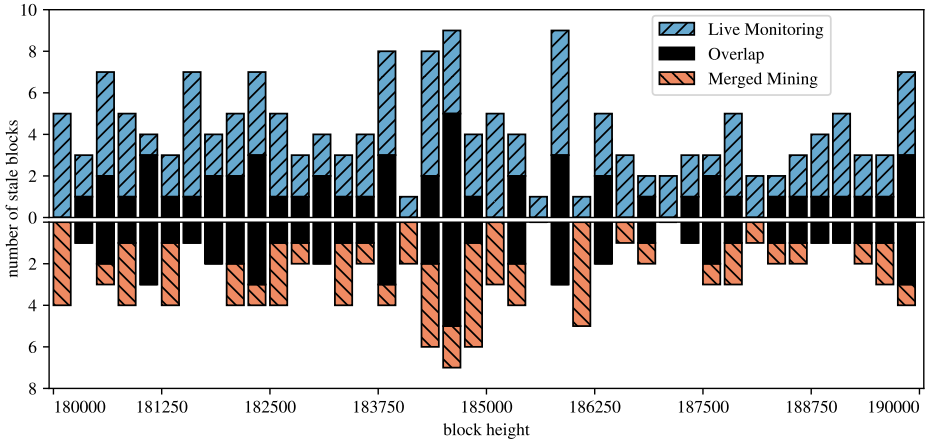

Fig. 3. Number of stale blocks observed by live monitoring (Decker and Wattenhofer) and merged mining; a single bar accounts for 250 blocks

Figure 3 shows our recovered stale blocks, the blocks captured by DW, as well as the overlap between the two data sets. All of the stale blocks we consider are linkable to the canonical parent chain and meet the correct difficulty requirement of the respective epoch, i.e. have a cryptographic link to Bitcoin. Surprisingly, our recovery technique is able to reveal 58 new stale blocks that were previously unaccounted for. The overlap in both data sets (54 blocks) further confirms that we are able to capture valid stale blocks observable through live monitoring. Combining both data sources, we distinguish 227 forks, which corresponds to a total fork rate of 2.27% for the first monitoring interval.

5.2 Long Range Comparison of Stale Blocks and Forks

Based on the initial approach from the DW monitoring interval, we extend our analysis over a wider time frame that stretches over the entire set of complete difficulty epochs for which we can recover stale blocks through merged mining, starting at epoch 74 and ending with epoch 264. Hereby, we aggregate and filter duplicates from all considered live monitoring data sources and compare the results to the stale blocks we were able to recover. Analogous to the methodology previously used, we only include stale blocks from our data that we can directly link to Bitcoin, i.e. are *not orphans*, and for which we can ensure that the PoW meets the target requirement of the parent chain at that height.

The results are shown in Figure 4, which contains some interesting patterns. First, an overall decline in the stale block rate as time progresses can be observed, which is to be expected as both, relay networks such as Falcon [3] (2016-06-8) and FIBRE [40] (2016-07-07), as well asand more efficient block announcement mechanisms, i.e. BIP130 [41] (2016-03-17), have come into play. Second, it appears that even though the overall stale block rate improves over time, blocks continue to be uncovered through merged mining which have otherwise not been observed.

In Figures 5 and 6 we further visualize this aspect by plotting, on the one hand, the derived *fork rate* and on the other hand, the ratio of stale blocks that were exclusively identified through merged mining. The latter also includes the ratio if we were to additionally consider orphan and shadow blocks that we link to an appropriate difficulty epoch and which would meet the prescribed difficulty requirement.

We further derive two *average total fork rates* for the Bitcoin network, including both live monitoring and merged mining, for difficulty epochs 146 to 209 and 209 to 264. The first range is chosen such that it begins with several of our live monitoring data sets and avoids gaps, while the second interval begins roughly after the commencement of relay network activities. Our results show a total fork rate of 0.85% for the first range of epochs (approx. 03/2014 – 07/2016) and 0.24% for the second range of epochs (approx. 07/2016 – 07/2018).

Based on our data, one possible explanation for the more recent increase in exclusively observed stale blocks, while at the same time observing a decrease in the fork rate may be, that the technique of fork observation through merged mining captures blocks which are either never announced over the p2p network or are not propagated for other reasons. The occurrence of such blocks would hence not be readily affected by improvements in the communication infrastructure and may stay at a certain level, even if the remaining fork rate is lowered.

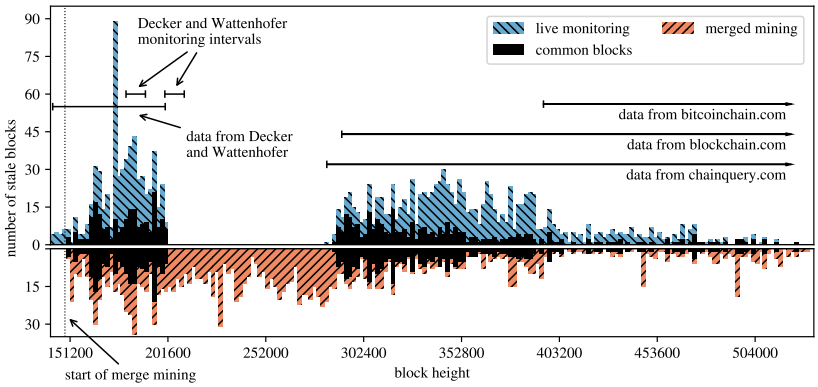

Fig. 4. Number of stale blocks observed by live monitoring (all considered data sources) and merged mining; a single bar accounts for a single difficulty period of 2016 blocks

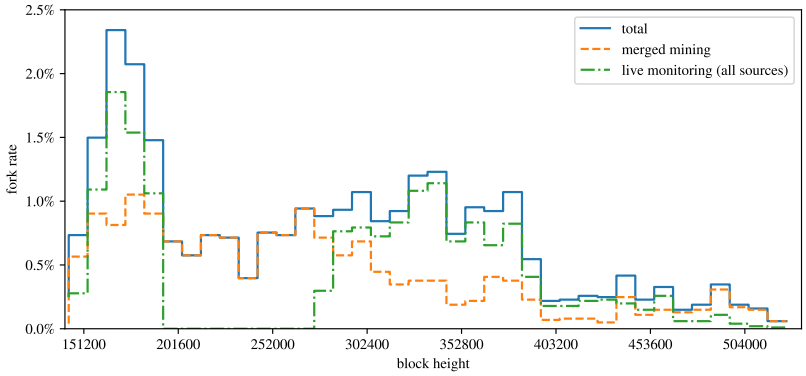

Fig. 5. Estimate of the fork rate in Bitcoin based on different data sources; 5 difficulty epochs grouped together

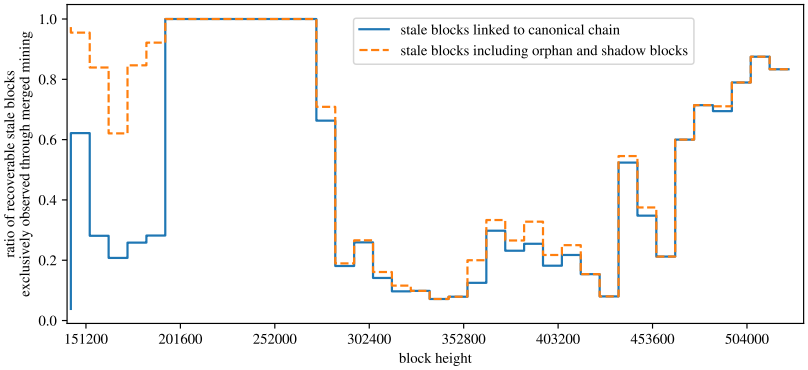

Fig. 6. Ratio of stale blocks exclusively identified through merged mining; 5 difficulty epochs grouped together

5.3 Stale Blocks and Forks in Litecoin

Through a stale block and fork rate recovery in Bitcoin we have shown the feasibility of our approach. By employing the same technique as before to Litecoin, we highlight that the same methodology can also be expanded to other merge mined blockchains. While we were able to obtain some live monitoring data [26] for Litecoin that includes forks, it is substantially less than what we were able to source for Bitcoin and not representative. We hence only show our recovered number of stale blocks as well as the fork rate, and point the interested reader to the appendix for further details (A.3). The methodology for deriving these values is analogous to the one previously used for Bitcoin.

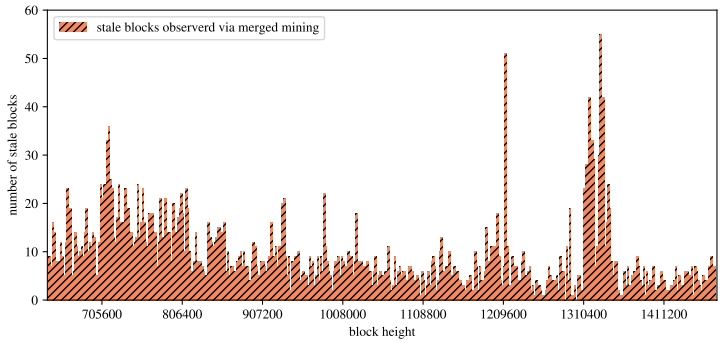

Fig. 7. Estimate of the stale block rate in Litecoin based only on merge mined data; 5 difficulty epochs grouped together

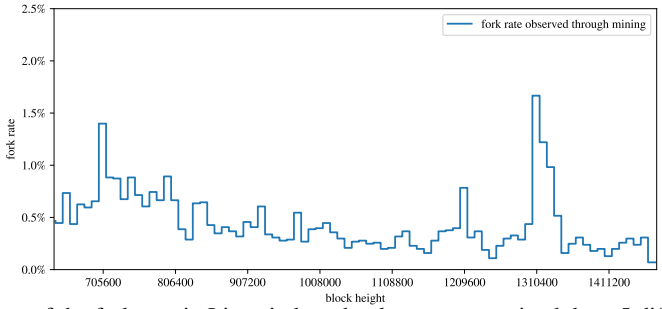

Fig. 8. Estimate of the fork rate in Litecoin based only on merge mined data; 5 difficulty epochs grouped together

6 Discussion

In Section 5 we show that the recovery of stale blocks and forks in merge mined parent cryptocurrencies is not only possible, but can also reveal new insights regarding their occurrence and the apparent inability to be fully captured by live monitoring alone. This raises the interesting question if other key metrics, such as the block propagation delay could, at least to some degree, be recovered in a similar fashion. Furthermore, it is important to acknowledge inherent limitations and necessary assumptions of this analysis method. Hereinafter, we address both of these points and additionally share some interesting anomalies and patterns that we discovered.

6.1 Recovery of other Metrics and Information

The side channel that is established to a merge mined parent may be useful beyond the recovery of stale and near block headers. In the following we present a list of possible application scenarios that we believe could be worthy of further investigation.

- **Block Time Estimates:** The time stamp encoded in Bitcoin block headers does not have to follow strict clock synchronization rules and hence can be relatively unreliable [1, 42]. Through merged mining, it may be possible to gain access to better timing information, both from the child block as well as the additional data from AuxPoW near blocks. In particular, merge mined cryptocurrencies with very short block intervals ranging in the order of seconds, such as GeistGeld [43], could prove helpful for improving timing estimates.
- **Information Propagation:** also relating to better time estimates, near and stale blocks recovered from the AuxPoW can provide additional information about a miner's particular choice which chain tip they were extending at that time. While it seems unlikely that a high enough sampling rate and timing precision through AuxPoWs can be achieved to reconstruct propagation delays, anomalies or large discrepancies may nevertheless be detectable.
- **Hash Rate Estimates:** The additional PoW samples that are available through AuxPoWs could help improve the quality and granularity of hash rate estimates [44] and may also allow for better approximations of how much of the computational power was split-off during past fork events.
- **Miner Behavior Analysis:** In Section 3 we have outlined that miner identification schemes are also applicable to AuxPoWs, because the full coinbase transaction is included for verification purposes. Thereby, the additional data gained from merged mining allows for a more detailed analysis of miner behavior, and may even reveal suspicious or adversarial behavior of bad actors, such as block withholding, if they also engaged in merged mining at the time.

6.2 Limitations

While merged mining can be used to recover certain information related to the parent chain, several limiting factors apply that may diminish its effectiveness. First and foremost, the presented technique only applies to currencies that have served as the parent in a merge mined relationship. The merge mining landscape is currently not well documented and information pertaining to merged mining in general is not readily available. Previous literature has shed some light on this topic [27, 28], however many details still remain relatively unknown outside of the specific mining communities. While we are aware of merge mining activities for cryptocurrencies that use Proofs-of-Work other than the herein considered DSHA256 and scrypt, such as X11 [45] or CryptoNote [46], we leave a detailed survey of potential merge mined parents to future work.

Furthermore, the effectiveness of merged mining as a side channel is dependent on a variety of factors, such as the degree of its adoption, the number and concrete parametrizations of the child chains, as well as the technique by which merged mining is achieved. In particular, the recovery of full block headers that meet the parent chain difficulty becomes increasingly unlikely, if only a small subset of the total hash rate is

actively participating in merged mining. A similar situation can be observed for long consecutive forks, where linking may be prevented if only shadow blocks are registered.

Another important issue is the fact that merge mined cryptocurrencies may have more than one possible parent with which they can be merge mined. Without an explicit cryptographic link to the canonical parent chain, orphan stale blocks could therefore belong to a different parent. For instance, we have recovered close to 15000 AuxPoW block headers that meet the encoded parent difficulty in their header, but which actually belong to a different parent cryptocurrency than Bitcoin (see appendix Table 1). Section 4 outlines how orphan stale blocks can be linked to a particular difficulty epoch in the analyzed parent, which discards these blocks as false positives as long as the difficulties of both parents are not the same. Nevertheless, certainty is only achieved when all orphan stale blocks are not taken into consideration. In our analysis in Section 5 we clearly state when such orphan stale blocks were included in figures or tables. Furthermore, we specifically decided not to rely on the additional data sources from live monitoring in our recovery process, which could have aided in bridging gaps between orphan stale blocks, to retain a clear picture of what is achievable solely through blockchain data and merged mining AuxPoWs.

6.3 Anomalies and Interesting Patterns

During the process of our analysis we were able to identify interesting patterns and anomalies that are not always readily explained by rational miner behavior. For instance, it is not widespread knowledge that several of the merge mined children to Bitcoin and Litecoin have, on occasion, also served the role as a *parent* for their siblings. This is possible because the requirements extended toward the AuxPoW do not include additional verification logic and only demand that the data structure and PoW follows an expected format. Because the child chains will generally also adhere to the same header format as the parent, it hence becomes possible to merge mine them interchangeably. As an example, the AuxPoW and parent block header of canonical Namecoin block "a10e863165101af92314…" at height 19236 actually stems from GeistGeld block "00000000000026c050e6…" at height 144590. We were further able to verify this insight through online references from respective mining communities [47, 48].

Another highly interesting pattern emerges when searching for the most concurrent forks extending a single parent block. For Bitcoin, we were able to identify a maximum of 18 concurrent stale blocks, i.e. 19 forks, that meet the correct parent difficulty and extend block "0000000000000d331567…" at height 153210. By applying a miner identification scheme similar to [28], we believe that all of these stale blocks were possibly mined by the same entity, namely *BTC Guild*. A similar pattern can also be observed in Litecoin, where the number of concurrent forks extending a parent is even higher, at a staggering 47 potentially valid blocks. In this case we were unable to achieve a possible match with a mining entity. The occurrence of such large concurrent fork events however is very rare in our recovered data, and the total count of situations where forks with more than 2 children exist is only 14 for Bitcoin and 37 for Litecoin. We believe that the above pattern can be explained by considering software issues or misconfiguration in the merged mining setup of the respective miner.

When checking for monotonically increasing heights, we were able to identify one case in Bitcoin where the BIP34 encoded height was not properly incremented while the

target difficulty of the canonical chain was met, rendering the block invalid. Overall, we discovered 55 blocks in Bitcoin and one block in Litecoin in our recovered data where the BIP34 encoded height did not correspond to the respective position to which the block could be linked in the parent chain and which are consequently invalid.

7 Related Work

In [1] Decker and Wattenhofer consider peer-to-peer network and information propagation characteristics of the Bitcoin network. Donet et al. [49] present an extensive survey of the Bitcoin p2p network and its topology, including block and transaction propagation delays, however information on stale blocks and forks is not included. Gervais et al. [2] presents a framework for quantifying and analyzing parametrizations of PoW blockchains. They include live measurements in Bitcoin, Litecoin, Dogecoin and Ethereum conducted in February 2016, from which a stale block rate of 0.41% for Bitcoin and 0.273% for Litecoin is derived. The presented simulation results predict a stale block rate of 0.14% (relay network and unsolicited block push) and 1.85% (standard propagation mechanism) for Bitcoin and 0.24% (standard propagation) for Litecoin. In Gencer et al. [3] a measurement study on decentralization metrics in Bitcoin and Ethereum is conducted, including aspects related to the peer-to-peer network such as provisioned bandwidth and latency. The work introduces a *fairness* property, which is defined as the ratio of a miner's share of pruned (stale) blocks to her mining power.

The encoding of additional data within the Bitcoin blockchain is addressed by Bartoletti and Pompianu [50], which analyze *OP_RETURN* metadata, and Matzutt et al. [51], which systematically analyzes and conducts an extensive quantitative and qualitative study of arbitrary encoded data within Bitcoin. Grundmann et al. [52] exploit characteristics of transaction processing and forwarding in the Bitcoin p2p network to infer its topology.

8 Conclusion

In this paper we outline and analyze a novel technique for recovering stale blocks through data that is accrued from merged mining. Thereby, we show that merged mining can act as a side channel for gaining information about the parent cryptocurrency, and that this data helps to infer key metrics such as the *fork rate*. Interestingly, a cryptocurrency is not trivially able to prevent another cryptocurrency from designating it as its parent in a merge mine relationship [28], and this fact has been identified as a potential attack vector in the context of hostile blockchain forks [53].

Our results indicate that live monitoring alone is not sufficient to capture all stale blocks and forking events in Bitcoin, as merged mining data is able to exclusively identify a majority of the stale blocks in more recent difficulty epochs. The authenticity of the recovered blocks and forks is hereby cryptographically ensured both, by the ability to link them to the canonical main chain, as well as the correct Proof-of-Work difficulty they satisfy. Important questions are therefore raised as to the nature of these newly identified stale blocks, to be addressed in future work.

Overall, we show that data embedded through merged mining can provide interesting new insights and may help augment and improve the fidelity and quality of empirical measurements to provide a more effective basis for future models, analysis, and simulations of Proof-of-Work cryptocurrencies.

References

1. C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*. IEEE, 2013, pp. 1–10. [Online]. Available: https://www.tik.ee.ethz.ch/file/49318d3f56c1d525aabf7fda78b23fc0/P2P2013_041.pdf
2. A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdo rf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC*. ACM, 2016, pp. 3–16.
3. A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer, "Decentralization in bitcoin and ethereum networks," in *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC). Springer*, 2018. [Online]. Available: http://fc18.ifca.ai/preproceedings/75.pdf
4. I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*. Springer, 2014, pp. 436–454. [Online]. Available: http://arxiv.org/pdf/1311.0243
5. 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. [Online]. Available: http://eprint.iacr.org/2015/796.pdf
6. A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," 2015, accessed: 2016-08-22. [Online]. Available: http://arxiv.org/pdf/1507.06183.pdf
7. J. Bonneau, "Why buy when you can rent? bribery attacks on bitcoin consensus," in *BITCOIN '16: Proceedings of the 3rd Workshop on Bitcoin and Blockchain Research*, February 2016. [Online]. Available: http://fc16.ifca.ai/bitcoin/papers/Bon16b.pdf
8. K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 264–279. [Online]. Available: http://www.cs.umd.edu/~jkatz/papers/whale-txs.pdf
9. P. McCorry, A. Hicks, and S. Meiklejohn, "Smart contracts for bribing miners," in *5th Workshop on Bitcoin and Blockchain Research, Financial Cryptography and Data Security 18 (FC). Springer*, 2018. [Online]. Available: http://fc18.ifca.ai/bitcoin/papers/bitcoin18-final14.pdf
10. A. Zamyatin, N. Stifter, A. Judmayer, P. Schindler, E. Weippl, and W. J. Knottebelt, "(Short Paper) A Wild Velvet Fork Appears! Inclusive Blockchain Protocol Changes in Practice," in *5th Workshop on Bitcoin and Blockchain Research, Financial Cryptography and Data Security 18 (FC). Springer*, 2018. [Online]. Available: https://eprint.iacr.org/2018/087.pdf
11. Blockchain.com, "Blockchain.com orphaned blocks," https://www.blockchain.com/btc/orphaned-blocks, Blockchain.com, accessed: 2018-09-25.
12. BitcoinChain.com, "Bitcoinchain bitcoin block explorer," https://bitcoinchain.com/block_explorer, BitcoinChain.com, accessed: 2018-09-25.
13. ChainQuery.com, "A web based interface to the bitcoin api json-rpc," http://chainquery.com/bitcoin-api, ChainQuery.com, accessed: 2018-09-25.
14. L. Project, "Litecoin," https://litecoin.org/, accessed: 2016-03-29.
15. Y. Sompolinsky and A. Zohar, "Accelerating bitcoin's transaction processing. fast money grows on trees, not chains," p. 881, 2013. [Online]. Available: http://eprint.iacr.org/2013/881.pdf
16. A. Miller and L. JJ, "Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin," 2014, accessed: 2016-03-09. [Online]. Available: https://socrates1024.s3.amazonaws.com/consensus.pdf
17. J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 281–310.

18. R. Pass and E. Shi, “Fruitchains: A fair blockchain,” 2016, accessed: 2016-11-08. [Online]. Available: http://eprint.iacr.org/2016/916.pdf
19. 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*. Springer, 2017, pp. 643–673.
20. K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, and E. Gün, “On scaling decentralized blockchains,” in *3rd Workshop on Bitcoin and Blockchain Research, Financial Cryptography 16*, 2016. [Online]. Available: http://www.tik.ee.ethz.ch/file/74bc987e6ab4a8478c04950616612f69/main.pdf
21. A. Kiayias and G. Panagiotakos, “On trees, chains and fast transactions in the blockchain.” 2016, accessed: 2017-02-06. [Online]. Available: http://eprint.iacr.org/2016/545.pdf
22. Y. Sompolinsky, Y. Lewenberg, and A. Zohar, “Spectre: A fast and scalable cryptocurrency protocol,” Cryptology ePrint Archive, Report 2016/1159, 2016, accessed: 2017-02-20. [Online]. Available: http://eprint.iacr.org/2016/1159.pdf
23. Y. Sompolinsky and A. Zohar, “Phantom: A scalable blockdag protocol,” Cryptology ePrint Archive, Report 2018/104, 2018, accessed:2018-01-31. [Online]. Available: https://eprint.iacr.org/2018/104.pdf
24. Bitcoin community, “Bitcoin-core source code,” https://github.com/bitcoin/bitcoin, accessed: 2018-09-25.
25. A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring, and B. Bhattacharjee, “Discovering bitcoin’s public topology and influential nodes,” May 2015, accsessed: 2016-03-09. [Online]. Available: http://cs.umd.edu/projects/coinscope/coinscope.pdf
26. chainz.cryptoid.info/, “Chainz blockchain explorers,” chainz.cryptoid.info/, chainz.cryptoid.info/, accessed: 2018-09-25.
27. A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, “Bitcoin and cryptocurrency technologies: a comprehensive introduction,” 2016.
28. A. Judmayer, A. Zamyatin, N. Stifter, A. G. Voyiatzis, and E. Weippl, “Merged mining: Curse or cure?” in *CBT’17: Proceedings of the International Workshop on Cryptocurrencies and Blockchain Technology*, Sep 2017. [Online]. Available: https://eprint.iacr.org/2017/791.pdf
29. M. Jakobsson and A. Juels, “Proofs of work and bread pudding protocols,” in *Secure Information Networks*. Springer, 1999, pp. 258–272. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-0-387-35568-9_18.pdf
30. 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, and Trust*, 2017.
31. A. Kiayias, A. Miller, and D. Zindros, “Non-interactive proofs of proof-of-work,” Cryptology ePrint Archive, Report 2017/963, 2017, accessed:2017-10-03. [Online]. Available: https://eprint.iacr.org/2017/963.pdf
32. Namecoin community, “Namecoin source code - chainparams.cpp,” https://github.com/namecoin/namecoin-core/blob/fdfb20fc263a72acc2a3c460b56b64245c1bedcb/src/chainparams.cpp#L123, accessed: 2018-09-25.
33. ——, “Namecoin source code - auxpow.cpp,” https://github.com/namecoin/namecoin-core/blob/fdfb20fc263a72acc2a3c460b56b64245c1bedcb/src/auxpow.cpp#L177-L200, accessed: 2018-09-25.
34. I0Coin community, “I0coin source code,” https://github.com/domob1812/i0coin, accessed: 2018-09-25.
35. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” Dec 2008, accessed: 2015-07-01. [Online]. Available: https://bitcoin.org/bitcoin.pdf
36. N. T. Courtois and L. Bahack, “On subversive miner strategies and block withholding attack in bitcoin digital currency,” arXiv preprint arXiv:1402.1718, 2014, accessed: 2016-07-04. [Online]. Available: https://arxiv.org/pdf/1402.1718.pdf

37. J. Göbel, P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: the selfish-mine strategy in the presence of propagation delay," 2015, accessed: 2015-03-01. [Online]. Available: http://arxiv.org/pdf/1505.05343.pdf
38. N. Developers, "Neo4j," https://neo4j.com/, 2012.
39. Gavin Andresen, "Bitcoin improvement proposal 34 (bip34): Block v2, height in coinbase," https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki, accessed: 2018-09-25. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki
40. Matt Corello, "Fast internet bitcoin relay engine," http://bitcoinfibre.org/, accessed: 2018-09-25. [Online]. Available: http://bitcoinfibre.org/
41. Suhas Daftuar, "sendheaders message," https://github.com/bitcoin/bips/wiki/Comments:BIP-0130, accessed: 2018-09-25. [Online]. Available: https://github.com/bitcoin/bips/wiki/Comments:BIP-0130
42. R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Block arrivals in the bitcoin blockchain," 2018. [Online]. Available: https://arxiv.org/pdf/1801.07447.pdf
43. GeistGeld community, "Geistgeld source code," https://github.com/Lolcust/GeistGeld, accessed: 2018-09-25.
44. A. P. Ozisik, G. Bissias, and B. Levine, "Estimation of miner hash rates and consensus on blockchains," arXiv preprint arXiv:1707.00082, 2017, accessed:2017-09-25. [Online]. Available: https://arxiv.org/pdf/1707.00082.pdf
45. E. Duffield and D. Diaz, "Dash: A payments-focused cryptocurrency," https://github.com/dashpay/dash/wiki/Whitepaper, Aug 2013, accessed: 2018-09-25. [Online]. Available: https://github.com/dashpay/dash/wiki/Whitepaper
46. N. Van Saberhagen, "Cryptonote v 2.0," https://cryptonote.org/whitepaper.pdf, Oct 2013. [Online]. Available: https://cryptonote.org/whitepaper.pdf
47. G. Hall, "Guide: Merge mining 6 scrypt coins at full hashpower, simultaneously," https://www.ccn.com/guide-simultaneously-mining-5-scrypt-coins-full-hashpower/, Apr 2014, accessed: 2018-09-25. [Online]. Available: https://www.ccn.com/guide-simultaneously-mining-5-scrypt-coins-full-hashpower/
48. united-scrypt coin, "[ann][usc] first merged minable scryptcoin unitedscryptcoin," https://bitcointalk.org/index.php?topic=353688.0, Nov 2013, accessed: 2018-09-25. [Online]. Available: https://bitcointalk.org/index.php?topic=353688.0
49. J. A. D. Donet, C. Pérez-Sola, and J. Herrera-Joancomartí, "The bitcoin p2p network," in *Financial Cryptography and Data Security*. Springer, 2014, pp. 87–102. [Online]. Available: http://fc14.ifca.ai/bitcoin/papers/bitcoin14_submission_3.pdf
50. M. Bartoletti and L. Pompianu, "An analysis of bitcoin op_return metadata," 2017, accessed: 2017-03-09. [Online]. Available: https://arxiv.org/pdf/1702.01024.pdf
51. R. Matzutt, J. Hiller, M. Henze, J. H. Ziegeldorf, D. Müllmann, O. Hohlfeld, and K. Wehrle, "A quantitative analysis of the impact of arbitrary blockchain content on bitcoin," in *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC). Springer*, 2018. [Online]. Available: http://fc18.ifca.ai/preproceedings/6.pdf
52. M. Grundmann, T. Neudecker, and H. Hartenstein, "Exploiting transaction accumulation and double spends for topology inference in bitcoin," in *5th Workshop on Bitcoin and Blockchain Research, Financial Cryptography and Data Security 18 (FC). Springer*, 2018. [Online]. Available: http://fc18.ifca.ai/bitcoin/papers/bitcoin18-final10.pdf
53. A. Judmayer, N. Stifter, P. Schindler, and E. Weippl, "Pitchforks in cryptocurrencies: Enforcing rule changes through offensive forking- and consensus techniques (short paper)," in *CBT'18: Proceedings of the International Workshop on Cryptocurrencies and Blockchain Technology*, Sep 2018. [Online]. Available: https://www.sba-research.org/wp-content/uploads/2018/09/judmayer2018pitchfork_2018-09-05.pdf

A Appendix

A.1 Considered Cryptocurrencies

Bitcoin PoW Family

- **Bitcoin** (BTC) We consider the sequence of blocks up to, and including, the block at height 532485 as Bitcoin's canonical main chain.
- **Bitcoin Cash** (BCH) While not well documented, Bitcoin Cash has also served as a parent cryptocurrency for merged mining. To help isolate and identify shadow blocks and orphan blocks that actually belong to BCH, we consider the canonical main chain from this fork ranging from block at height 478559 to block height 544355.
- **Namecoin** (NMC) was the first cryptocurrency to implement merged mining, starting at its own block height of 19200 and corresponding to a point in time after Bitcoin block at height 148553, which sets the commencement date of merged mining to 08.10.2011. Just like Bitcoin, NMC has a target block interval of 10 minutes. We consider Namecoin blocks up to a block height of 409629.
- **IXCoin** (IXC) is a fork of the Bitcoin protocol that was publicly announced in August 2011. IXCoin has the same target block interval as BTC and NMC (10 minutes) and commenced merged mining at a block height of 45001 on December 31, 2011. We include information up to block 455051.
- **I0Coin** (I0C) is a clone of the Bitcoin protocol that reduces the target block interval to 90 seconds and re-targets the PoW difficulty every 120 blocks. It was launched on August 15, 2011 and commenced merged mining at a block height of 160045 on December 20, 2011. We use information from I0Coin (I0C) up to block 2556904.
- **GeistGeld** (XGG) is an experimental protocol fork of Bitcoin that tries to test out the limits of block generation rate with a target block interval of 15 seconds. It was reported that GeistGeld fell victim to a time warp attack in September 2011 [6] which would render the time stamps in the child blocks unreliable for the attack period, however we do not rely on such time stamps for our analysis. XGG commenced merged mining at block height 14092 and we consider blocks up to height 7309971. We had to modify the available source code to expose the necessary AuxPoW information through the RPC interface.
- **Devcoin** (DVC) has the goal of helping to fund open source projects. It is a protocol fork of Bitcoin and has a target block interval of 10 minutes and difficulty adjustment after 144 blocks and merged mining started at block 25000. We use information from Devcoin up to block 337624.
- **Groupcoin** (GPC) is similar to DVC in its intention to donate a portion of the mined coins for open source developers and writers and was announced in June 2011. Groupcoin is not to be confused with an equally named Proof-of-Stake coin presented in 2014. It has a difficulty adjustment period of 2016 blocks and a target block interval of 10 minutes. Merged mining commenced at block height 17187 and we consider blocks up to height 235751.

[6] See `https://bitcointalk.org/index.php?topic=43692.msg521772#msg521772`

– **Unobtanium** (UNO) was launched in October 2013 and is a fork of the Bitcoin code base. It has a target block interval of 180 seconds and merge mining was started in May 2015 at block height 600135. We consider Unobtanium blocks up to a height of 1163483.

Litecoin PoW Family

– **Litecoin** (LTC) is a fork of Bitcoin launched in 2011, which replaces DSHA256 with scrypt as PoW algorithm, in an attempt to counter the ASIC-dominated mining. It maintains a target block interval of 150 seconds, a difficulty adjustment period of 2016 blocks and acts as a parent for scrypt-based merged mining. We consider blocks up to a height of 1477146.
– **Dogecoin** (DOGE) initially was launched as a non-serious project based on an internet meme in 2013, however was able to attract and maintain a vivid community. It is roughly based on the Litecoin codebase, maintains a target block interval of 60 seconds and adjusts difficulty after every block. It was the first cryptocurrency to introduce scrypt-based merged mining (at block height 371337). We consider blocks up until 2357918.
– **Viacoin** (VIA) is a fork of Bitcoin using scrypt as PoW algorithm, launched i.a. as reaction to the (temporary) reduction of Bitcoin's `OP_RETURN` size 2014. It maintains a target block time of 24 seconds and recalculates the difficulty target after every block. The first merged mined block was generated at height 551885. We consider blocks up until block height 5324736.

Cryptocurrency	PoW	Merge M.	Start of Merge M.	Considered Block Heights	Parent Blocks	Child Blocks
Bitcoin (BTC)	DSHA256	✗	-	0 – 532485	181658	0
Bitcoin Cash (BCH)	DSHA256	✗	-	478559 – 544355	12389	0
Namecoin (NMC)	DSHA256	✓	19200 (2011-10-08)	0 – 409629	0	390300
IXCoin (IXC)	DSHA256	✓	45001 (2011-12-31)	0 – 455051	861	409969
I0Coin (I0C)	DSHA256	✓	160045 (2011-12-20)	0 – 2556904	1620	2395170
GeistGeld (XGG)	DSHA256	✓	14092 (2011-09-16)	0 – 7309971	2	2493631
Devcoin (DVC)	DSHA256	✓	25000 (2012-01-07)	0 – 337624	135	312624
Groupcoin (GPC)	DSHA256	✓	17187 (2012-02-16)	0 – 235751	0	218494
Unobtanium (UNO)	DSHA256	✓	600135 (2015-05-08)	0 – 1163483	6	561355
Litecoin (LTC)	scrypt	✗	-	0 – 1477146	699714	0
Dogecoin (DOGE)	scrypt	✓	371337 (2014-09-11)	0 – 2357918	3	1983945
Viacoin (VIA)	scrypt	✓	551885 (2014-12-25)	0 – 5324736	973	4767508

Table 1. Considered Blockchain Data of Merge Mined Cryptocurrencies and their Parents

Cryptocurrency	Source	First Block Height	Start Time	Last Block Height	Stop Time	Stale Blocks
Bitcoin	Decker and Wattenhofer [1]	142258	2011-08-23	200206	2012-09-23	612
Bitcoin	blockchain.com	291123	2014-03-18	525890	2018-06-04	932
Bitcoin	chainquery.com	283421	2014-01-31	525890	2018-06-04	715
Bitcoin	bitcoinchain.com	395001	2016-01-25	525890	2018-06-04	51
Litecoin	chainz.cryptoid.info	1217073	2017-06-05	1472513	2018-08-11	223

Table 2. Considered Live Monitoring Data for Bitcoin and Litecoin

A.2 Bitcoin Total Number of Stale Blocks for Different Data Sources

Table 3 shows both, the total number of unique stale blocks exclusive to the data source, as well as the overall number of (non-duplicate) stale blocks it contains.

Table 3. Comparison of total stale blocks in Bitcoin observed by different live monitoring sources and merged mining

	unique stale blocks	total stale blocks
Merged Mining	1164	1678
Decker and Wattenhofer	410	612
blockchain.com	256	932
chainquery.com	113	715
bitcoinchain.com	4	51
	2863	3988

A.3 Litecoin Stale Block Rate Comparison

As we have previously outlined in Subsection 5.3, the live monitoring data we were able to obtain for Litecoin was relatively limited and only contained 223 stale blocks/forks. Nevertheless, we plot this live monitoring data against the recovered stale blocks through merged mining in Figure 9 and show that the data sets also contain some overlap. Again, our recovered data only contains stale blocks that can be cryptographically linked to the canonical Litecoin chain and which meet the prescribed difficulty target.

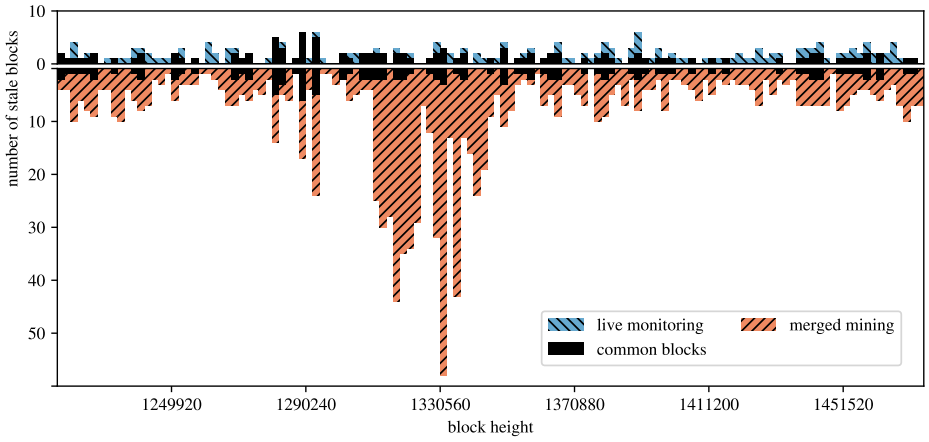

Fig. 9. Stale block rate recovered from merged mining in Litecoin compared to available live measurements [26]; 5 difficulty epochs grouped together

Table 4. Structure of the coinbase of a merge-mined block [30]

Field name		**Type (Size)**	**Description**
coinbaseLen		VarInt (1-9 bytes)	Length of the `coinbase` field in bytes as a variable length integer. Maximum size is 100 bytes.
coinb.	blockHeightLen	(1 bytes)	Length in bytes required to represent the current `blockHeight`.
	blockHeight	(3 bytes)	Current block height.
	[data]	char[] (0-52 bytes)	Optional: Arbitrary data that can be filled by the miner (e.g., identifying the block miner)
	[magic]	char[] (4 bytes)	Optional: If *len(coinbase)* $\geq$ 20, `magic` bytes indicate the start of the merged mining information, e.g., "\xfa\xbe\x6d\x6d"
	BlockHash or MerkleRoot	char[] (32 bytes)	Hash of the merge-mined block header. If more than one cryptocurrencies are merge-mined, this is the Merkle tree root hash of those cryptocurrencies.
	MerkleSize	uint32_t (4 bytes)	Size of the Merkle tree, i.e., the maximum number of contained cryptocurrencies.
	MerkleNonce	uint32_t (4 bytes)	Used to calculate the indices of the mined cryptocurrencies in the Merkle tree. If no Merkle tree is used, it is set to 0.