POLITECNICO DI TORINO

Master degree course in ICT FOR SMART SOCIETIES

Master Degree Thesis

# Decentralizazion Problems and Cross-Pool Mining in Bitcoin

**Supervisor:**
prof. Carla Fabiana Chiasserini

**Candidate**
Matteo Romiti
matricola: 225907

**Internship Tutor**
Bernhard Haslhofer

Academic Year 2017-2018

# Summary

The current state of mining power distribution among mining pools resembles an oligopoly, where six entities control more than 75% of the market. With this work, we intend to assess the level of decentralization of bitcoin mining and its evolution as it is a fundamental aspect of the stability and the security of the peer-to-peer protocol. Also, little is known about the economic relationships of these mining pools, their payout schemes and their members. We discovered that since the beginning of 2018, a single company has been in control of a safety-critical level of mining power. We also noticed that mining pools are mining behind other pools, either to make their revenues steadier or to conceal their real mining power. Finally, we tried to reveal identities of pools' members and their shares within a mining pool. Our extracted data about payout addresses and mining entities will be publicly available in order to improve the knowledge about the current bitcoin mining landscape.

**French Version** Le minage des bitcoins est devenu un marché très compétitif, où une poignée d'entreprises valide la vaste majorité des transitions. Nous montrons comment la centralisation actuelle de la puissance de minage représente une menace pour la sécurité des protocoles, Comment les pools de minage partagent des récompenses de bloc et comment ces entités sont intimement liées. Nous poursuivrons sur qui sont ces membres du pool et quel est leur comportement.

Toutes nos données et informations extraites sur l'activité de minage (blocs, adresses et piscines) seront disponibles publiquement pour stimuler la recherche dans le domaine et identifier des possibles menaces.

# Acknowledgements

I want to thank all the people that contributed to my education.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation and Aim of the Work

Bitcoin mining has become a very competitive market in which a handful of players are validating the vast majority of transactions. Already in 2014, the protocol was under threat by a centralization of power into one single pool and its members agreed on shifting their hash rate away to other pools. In recent months, a similar issue is arising but under different conditions: a single entity is behind a large share of the mining industry and of the current hash rate. This is seen as a major concern in a peer-to-peer protocol like Bitcoin which has now attracted a lot of investors and created high expectations for people in countries without a solid and reliable banking system.

Miners have an incentive to conceal or obfuscate their real mining power: it would allow them to maximize market shares and profits without harming the security and credibility of the system that is the source of their revenue streams. A possible obfuscation strategy is to generate multiple identities, which are then used in different mining pools.

In order to shed light on Bitcoin's real-world mining power, we conduct an empirical analysis of the market share of mining pools in order to confirm or reveal economic relationships between miners and mining pools. We do this by investigating the flow of mining rewards from the coinbase down to the level of individual miners, with possible distribution schemes in between. We show how the current centralization of mining power is a threat to the protocol's security, how mining pools share block rewards and how these entities are related to each other. We further develop on who are the pools members and what's their behavior. All our data and extracted information on mining activity (blocks, addresses and pools) will be publicly available to foster research on the field and identify possible threats.

# Chapter 2

# Background

## 2.1 Blocks, Transactions and Mining

### 2.1.1 Blocks and Transactions

Since January 2009, a set of computers have been communicating to each other through a peer to peer protocol called Bitcoin. The purpose of this network is to create roughly every ten minutes a new *block*. A block is a data structure with a header, an integer value representing the number of transactions in the block and a list of transactions where parties exchange a digital currency called bitcoin (BTC). The header of a block contains:

- the *version number* to indicate which set of validation rules to follow,
- the *previous block hash*, the result of hashing twice the header of the previous block header using SHA256,
- the *merkle root* derived from the hashes of all transactions included in the block,
- the Unix *timestamp* when the miner started creating this block,
- an encoded version of a number called *target threshold*,
- the *nonce*, an arbitrary number miners change.

A transaction is another data structure that describes the exchange of bitcoin between one or more input addresses and one or more output addresses. An address is a Base58Check-encoded string derived from the public part of a public/private ECDSA keypair. An address can be generated through the Bitcoin core client [2] or through any wallet service and will have a public and a private key associated. When generated, an address has no funds, but it can receive any amount of bitcoin from any other address through a transaction. In order to spend any sum of BTC received by an address, its owner must provide the private key associated to that address. One address can receive different amounts of BTC from different transactions and they can be redeemed in one single transaction or in many distinct transactions. Output addresses of a transaction can be owned by different entities, meaning that their private keys are not controlled by the same person. Input addresses instead are owned by the same entity because it must provide all the private keys associated to the input addresses.

### 2.1.2 Mining

When two parties agree to perform a transaction to transfer digital coins from one address to another, they will use a specific software (i.e., wallet service). The sender must also agree to pay a fee to the Bitcoin network to incentivize nodes to validate the transaction and include it in a block. Once this happens, the transaction is sent to the peers to which both parties are connected. These peers will first validate the transaction according to a set of *consensus rules* established by the Bitcoin protocol, then they will re-transmit it to other peers and eventually all nodes in the network will receive this transaction. When a node in the network has received enough valid transactions, which total size is not greater than 1 MB, it will start looking for a valid nonce to include in the header. A valid nonce is a 4-byte string that turns the double SHA-256 hash of the block into a number lower than a target number. The lower the target number, the more difficult it will be to find a valid nonce. This difficulty parameter is updated every 2016 blocks (roughly two weeks) to keep the average time between two mined blocks around ten minutes.

One more fundamental aspect in this procedure of creating a block is the following. We now know that a node has the incentive to include as many transactions in a block as possible and then look for a valid nonce because of the fees inputs agreed to pay. Nevertheless, there is another incentive for a node to perform this work. Before looking for a valid nonce and after the validation of each transaction, a node is allowed to add one transaction to the block, still satisfying the 1 MB size limit. This transaction must be the first in the block and is called coinbase. In this transaction, the output can be any set of addresses known as *payout addresses* and chosen by the node creating the block while the input is not a standard address, rather an arbitrary string that the node can decide and finally the "exchanged" sum, usually referred to as *block reward*. This sum depends on two values that we will call $B$ and $F$. The first value $B$ corresponds to newly generated coins and it is halved every 210,000 blocks (approximately every four years), while the second value $F$ is the sum of the fees collected from the transactions included in the block. Obviously, when choosing the transactions to include in a block, a mining node will choose the transactions with the highest fees in order to maximize its revenues.

Since every block has a reference in its header to the previous block, this sequence of blocks can be seen as a singly linked list. Whenever a new block is created, it is appended to the head of the list. The first block ever created in January 2009 is called *Genesis Block*. This list of linked blocks is generally known as block chain. For an easier reference, each block is also associated to an integer number representing its height. The Genesis Block has height 0, while any other mined block has a height equal to the height of the previous block plus 1.

The process of creating a block with valid transactions, adding the coinbase transaction and then looking for a valid nonce is called mining and, more specifically, it is a Proof-of-Work algorithm. This work is a race in that only the first node that succeeds in finding a valid nonce will be able to append the newly mined block to the chain and to receive the block reward. Once this happens, the winner will broadcast as fast as possible the new block to its peers and let the whole network verify whether its nonce actually is valid. After checking the validity of the block, a mining node performs two actions: re-transmitting the block to its peers and start mining a new block on top of the one just received, which means including the hash of its header into the new block header.

If there are $n$ miners with a mining power $m_i$, with $i = 1, 2, ..., n$, each miner finds a block with a time interval which is exponentially distributed with mean $m_i^{-1}$.

## 2.2 Decentralized Consensus

When nodes verify blocks and transactions using the consensus rules, they eventually agree on a single universal "truth". Eventually because it may happen that two nodes mine two valid but different blocks at the same time and start propagating their chains that differ only on the last block and this temporary situation is known as fork. Each node will always mine on the longest chain it is aware of, i.e., the one with the highest sum of difficulty values of all blocks. A fork will always be resolved into a winning chain, the main chain and a list of orphan blocks. Orphan blocks are valid blocks, but they are neglected by the network and the work done by their miners is not rewarded.

It is clear that having a well-connected node allows a miner to quickly broadcast its chain to the network and let the other peers validate and start mining on its work. This aspect plays an important role for a miner as its revenues depend not only on its hash rate, but also on its orphan rate as it is a measure of how much mining power is wasted.

## 2.3 Mining Hardware Evolution

The first block of the Bitcoin chain was created on January, 3rd 2009 and it contains only the coinbase transaction. A famous message was included in the input: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks", to emphasize the reasons behind the creation of this alternative payment method. For more than one year, there were only a few computers connected and using the peer-to-peer protocol and blocks were generated only with coinbase transactions.

Initially, mining meant running a piece of software on your machine and scan the entire search space of nonces in a linear manner using a standard CPU. Years later, when more people got involved in this digital currency, CPU mining was replaced by GPUs in order to parallelize the nonce search. Multiple GPUs were connected to one CPU and overclocked to increase profits. The drawback was that GPUs were not made for running side by side and overheating turned out to be a big issue. In 2011, FPGAs started appearing and performances and cooling improved but the hardware costs were higher. After few months, manufacturers started designing ASICs for bitcoin mining and they are now the most profitable hardware. Professional mining is now based on big farms filled with thousands of ASICs in regions with cheap energy, low temperatures and fast connection, probably not the decentralized vision Satoshi Nakamoto had in mind when creating the Bitcoin protocol.

## 2.4 Pooled Mining

Towards the end of 2010, pooled mining started. In this activity, a set of mining nodes decide to work together to find a valid nonce and mine the next block. The idea is that as more miners join a pool, the total hash rate of that pool will increase and, consequently, mining a block becomes faster. The reward from each new block must then be split among pool members according to some payout schemes that we will discuss in section 2.4.3. Compared to solo mining where one looks for a valid nonce only with its own hardware and gets the entire block reward, pooled mining is more attractive and more logical from an investment viewpoint because the payoffs, despite being smaller, are more frequent. In solo mining, creating a valid block might take years or decades and yield a lot of bitcoin, while in pooled mining, it is not difficult to get some millibitcoin every week.

The structure of a pool can be summarized as follows. One pool manager is connected to the P2P network to receive and validate transactions. Then, when enough transactions are gathered, it sends the block header to pool members and they will start searching for a valid nonce in parallel. If a member succeeds in this task before a new block is received by the pool manager, it submits its result and the pool will get the block reward, otherwise its work is wasted and a new block header will be created.

As already explained in section 2.1.2, the miner of a block can insert any arbitrary string in the coinbase input. To publicly claim mined blocks and to incentivize miners to join a pool, it is common to find in the coinbase some text related to pools or to mining software. We will refer to these strings as *coinbase tags.*

Moreover, a coinbase transaction of a block mined by a pool is usually containing one single output address. This address, known as payout address, will then split the block reward among pool members with what we call a *payout transaction.*

The first mining pool was Slush and it is still one of the major pools. Many other pools appeared and then lost momentum. Most of them were, or are, public and everyone can join, few of them are instead private. In July 2014, a pool named GHash managed to reach a mining power above 50% of the total mining power, meaning that on average more than one block every two were mined by one pool, raising alarms in the entire community. Soon afterwards, pool members moved away from GHash.io, which promised to limit its hashing power to 40%. At time of writing, the pools with highest hashing rate are BTC.com, AntPool, BTC.TOP, ViaBTC, F2Pool, and Slush.

One additional detail is that nothing prevents a mining hardware owner to point its hash-rate power to multiple pools simultaneously and get shares of block rewards from each of them. We will refer to this activity as *cross-pool mining.* It must be noted that it is actually a good practice because it increases the level of decentralization in the mining industry, which is currently a concern. On the other hand, it could also be seen as a way to hide a large amount of hashing power owned by one entity, a hash laundering mechanism [12].

### 2.4.1 Bitmain Controversy

Bitmain has become the leader in manufacturing bitcoin mining hardware and this is not a news anymore [16]. Other than producing mining rigs and selling them, the company is also active in the mining activity as well. It runs AntPool and in July 2016 it acquired Blocktrail, which then renamed BTC.com [3]. In April 2017, ViaBTC received $2.9M in a Series-A funding from Bitmain [4]. Despite being three different mining pools with a large number of members each, it is worth asking whether Bitcoin is still a decentralized protocol and crypto-currency. It is also known that a fraction of the mining hardware produced by Bitmain is used by the company itself to mine on its pools (AntPool in particular).

We will see more on this topic in section 3.3. The key factors that led Bitmain to be the leader are: low production costs thanks to cheap Chinese labor and hardware, access to cheap electricity and of course large mining farms to get the advantages of an economy of scale.

### 2.4.2 Coinbase Flow

One concept that must be introduced for a general understanding of this work is what we call a *coinbase flow*. When a block is mined, the newly minted coins are received by one or more payout addresses which can in turn transfer the sum to other addresses and the coins start being exchanged in the network. We will refer to this set of transactions and addresses as the coinbase flow. Each coinbase flow originates from a coinbase transaction that we call level zero. The outputs of a coinbase transaction, both the payout addresses and the bitcoin received, form the *first level* of the coinbase flow. If at least one output is spent, we will have a second level. This flow of money then continues to the next levels and the number of transactions and addresses involved in it increases quickly. This work focuses on two aspects of a coinbase flow: the first level, where we have payout addresses of mining pools, and few deeper levels where we expect to find payout transactions to pool members.

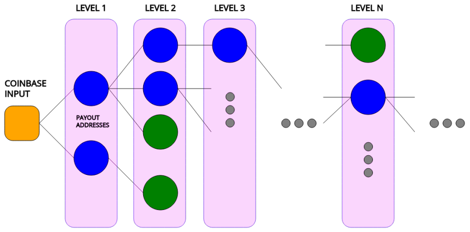


Figure 2.1. A simple coinbase flow. Nodes represent inputs and outputs of a transaction. Green nodes are unspent transaction outputs (UTXOs), while blue nodes are redeemed outputs.

### 2.4.3 Payout Schemes

Each pool is formed by many members, which perform mining work and claim their respective share of block reward. In this section we explain how a pool member gets paid and what are the schemes a pool manager can offer to miners. More details and other payout schemes can be found in [14] and in [15].

We know that the Bitcoin protocol sets a target for finding a valid nonce every 2016 blocks, the lower the target, the higher the difficulty of mining. A mining pool will set two thresholds: one is the target of the protocol itself and the other is an internal target. This internal threshold is higher than the official one which means finding a nonce below this threshold is easier for pool members. Each time a member finds a pool-valid nonce, i.e., below the internal threshold, she gets a share of the future block reward. Eventually, it will happen that when looking for a valid internal nonce, a member will find a nonce below the target of the protocol and the pool will be able to mine the block and get its reward to then split among members according to their shares. The higher the internal threshold, the larger the search space for a valid nonce, the easier to find it, the higher the number of members submitting their nonces and receiving a share of the block reward.

In mining, we can define:

- a difficulty $D$ so that every computed hash will lead to a valid block with probability $p \simeq \dfrac{1}{2^{32}D}$,
- a single miner with hash rate $h$ and a mining pool with hash rate $H$,
- a miner-to-pool hash rate ratio $q = h/H < 1$,
- a block reward $B$ and a pool manager keeping a fraction $f$ of the block reward,
- a miner submitting $n$ shares per round[1] to its pool with a total number of shares per round equal to $N$.
- pool members find and submit *shares*, hashes of a block header which are low enough to have made a block if the difficulty $D$ was 1 (lower than the real difficulty)

Then, we can state the following:

- finding a block mining solo (or in a pool) is a Poisson process and mining for time $t$ results in $\dfrac{ht}{2^{32}D}$ (or $\dfrac{Ht}{2^{32}D}$) blocks on average,
- pool mining variance is $q$ times the solo mining variance,
- expected payout for a miner in a pool is $\dfrac{(1-f)htB}{2^{32}D}$,
- each hash submitted by a pool member has a probability of $\dfrac{1}{2^{32}}$ to be a share,
- every share has a probability $p = 1/D$ to be a valid block,
- a miner in a pool should receive $(1-f)pB$ per share submitted,

[1] Time between one block found by the pool to the next.

- the number of shares in a round follow a geometric distribution and it's a memory-less process.

Depending on the payout scheme used by pools, mining can be more profitable if a miner changes pool at the specific points in time, leading those who mine continuously to earn less than their due reward. This is known as pool-hopping.

### Proportional

The reward per round per miner in a pool is $\dfrac{n(1-f)B}{2^{32}D \cdot N}$. A pool-hopper would leave the pool when there are $rD$ shares submitted, with $r = 0.435$, and join a pool with fewer submitted shares (there are $D$ submitted shares on average per round).

### PPS (pay-per-share)

Each share submitted by a pool member is worth $(1-f)pB$, thus a deterministic value known in advance. The operator takes all the risks: in a long round he would likely lose money, while in a short round he would be able to keep most of the block reward for himself. To compensate this risk the operator will charge higher fees. The method is hopping-proof.

In FPPS (full pay-per-share), a standard transaction fee within a certain period is computed and added to the block reward, then standard PPS is used.

### Slush and geometric method

In the method created by Slush pool, each submitted share is worth $s(1-f)B$, with a score $s = exp(T/C)$, where $T$ is the "age" of the share in the round and $C$ is a tunable parameter. Here, a steady state is reached only some time after the round start (it is more profitable to mine early) and there is no consideration of the current difficulty, hence the pool is hoppable based on expected difficulty adjustments.

To solve this issue, the geometric method introduces a fixed fee and a variable fee exponentially decaying with time so that there is always a steady state and the pool is not hoppable.

### PPLNS (pay-per-last-N-shares)

In this method, there is no concept of "rounds", rather members are paid considering the last N shares. A share is worth $\dfrac{(1-f)B}{N}$ if submitted in the last $N$ shares, but it is not hopping-proof if $D$ and $B$ change: a pool-hopper would leave when $D$ increases and join when it decreases. A hopping-proof variant is unit-PPLNS.

### Double Geometric method

In double geometric scheme instead round boundaries are crossed, but not ignored: Every block found reduces the reward to be given for future blocks, but does not erase it completely. A parameter controls the degree of reduction, thus setting the exact location on the spectrum between PPLNS and geometric.

## 2.5 Multiple-Input Clustering Heuristic

A transaction can have multiple inputs where each input can be seen as a pair of $(address, UTXO)$. $UTXO$ stands for unspent transaction output and it's the amount of bitcoin that can be redeemed by $address$.

Let's assume we find in the bitcoin blockchain a transaction $tx_1$ with two input addresses: $a_1, a_2$ and another transaction $tx_2$ with two input addresses $a_2, a_3$. Regardless of the output addresses of these two transactions, we can conclude that there is one single entity controlling the private keys of the addresses $a_1, a_2, a_3$, because $a_2$ appears as input in both transactions. More formally, given two sets of input addresses in two different transactions $A_{tx_1} = \{a_{1,1}, ..., a_{1,N}\}$ and $A_{tx_2} = \{a_{2,1}, ..., a_{2,M}\}$, we have a cluster if $A_{tx_1} \cap A_{tx_2} \neq \emptyset$. This technique is known as multiple-input clustering heuristic and has proven to be very effective in many cases A graphical representation is shown in Figure 2.2.

Nevertheless, it has some drawbacks in that it may happen that some transactions have as input lots of addresses, not necessarily owned by the same entity. This actually happened in the past due to the activity of big crypto-currency exchanges like the Japanese Mt.Gox, that led to the formation of the so-called *super-clusters* that include millions of addresses.

One of the most interesting aspects of this heuristic is that if we manage to reveal the identity behind one address in a cluster, we can automatically link all the other addresses in that cluster to the same identity. Revealing an identity usually means looking into forums, websites, previous research or also transacting directly with someone [11]. Once the association is established, we will be able to map a cluster and all its addresses to a *tag*.

The Austrian Institute of Technology (AIT) has its own implementation of this heuristic [10] and it allows us to quickly reveal the identity of many addresses. We will rely on this technique throughout the rest of this work and will play an important role.

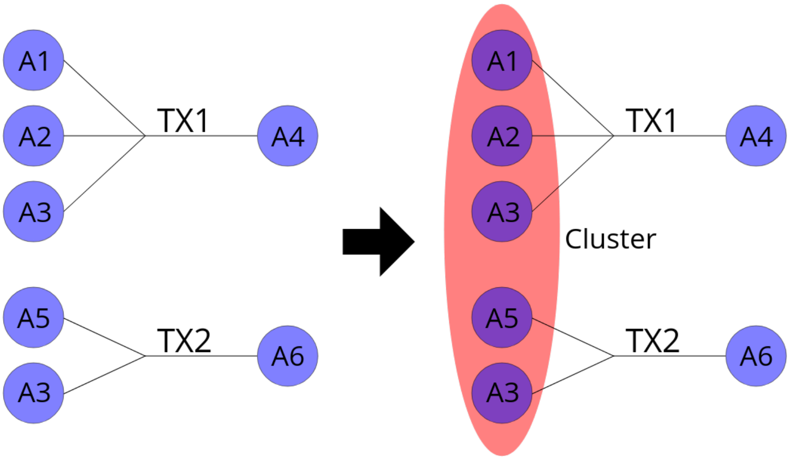


Figure 2.2. Multiple-input clustering heuristic. In Transactions with multiple inputs, the input addresses can be linked to one entity, represented by a cluster, owning their private keys. When an address, $A3$ in this case, is reused in another multiple-input transaction, we can group these two sets of addresses into one single cluster.

## 2.6 Related Work

### 2.6.1 Attacks on Mining

A miner with more than 50% of the total hash rate can only attack the network in future blocks or changing the most recent ones by double-spending bitcoin—only on the attacker's own transactions—or excluding some transactions or blocks from other miners. Nevertheless, it cannot steal bitcoin, change past transactions or ownership records.

#### Selfish and Stubborn Mining

Selfish mining [8] happens when a pool does not publish immediately a mined block and keeps mining on top of it, the forked chain is published when another block is mined by another pool and a race begins.

Let's assume $\alpha$ is the hash rate of the selfish miner, while the rest of the miners have a hash rate of$(1-\alpha)$. Let's also assume $\gamma$ is the ratio of honest [2] miners that

[2] This means they are not controlled by the attacker, but they are unaware of the origin of the

starts mining on the chain published by the attacker, while a fraction $(1-\gamma)$ starts mining on the other chain. I turns out that

- when $\gamma = \frac{1}{2}$, selfish mining is more profitable than normal mining if $\alpha \geq \frac{1}{4}$;
- when $\gamma = \frac{1}{4}$, selfish mining is more profitable than normal mining if $\alpha \geq \frac{3}{10}$;

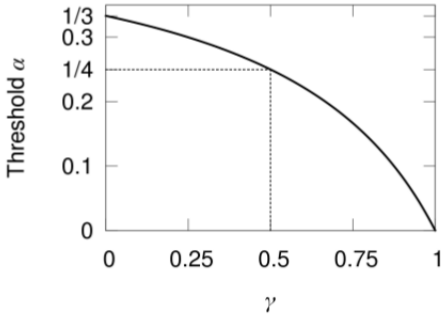


Figure 2.3. From [8], for a given $\gamma$, the threshold $\alpha$ is the minimum power a selfish mining pool need to be more profitable than normal mining.

In [13], Nayak et al. show that alternative mining strategy named "stubborn mining" can be more profitable than selfish mining for certain values of $\gamma$ and $\alpha$. Instead of stop mining on its private chain as soon as the public chain is longer, a pool can be stubborn and continue mining on its chain hoping that it will catch up with and surpass the public chain in the near future. The values of $\alpha$ and $\gamma$ for which stubborn mining is better than selfish mining are shown in Figure 2.4.

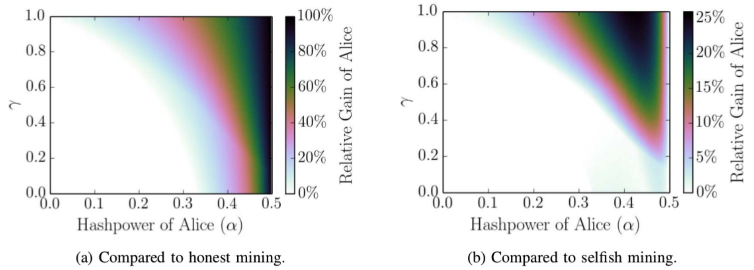
### 2.6.2 Mining Power Distribution

There are few websites[3] that are currently providing statistics about mining power distribution and a more in-depth analysis of mining pools is available in [1], but the last update was on November 2016 and recent developments in bitcoin mining are not covered.

In [9], the authors analyze different decentralization metrics of Bitcoin and Ethereum, among which we find the hash rate distribution. Unfortunately, this

blockchain they are mining on.

[3] blockchain.info/pools, btc.com/stats/pool, bitcoinchain.com/pools



(a) Compared to honest mining.

(b) Compared to selfish mining.

Figure 2.4. From [13], relative gain of stubborn mining for different values of $\gamma$ and $\alpha$.

study is limited to a 10-month period starting on July 15, 2016 and relying only on a single source of information, Blocktrail. As we will see in section 3.1.1, we will gather data also from other sources and considering a longer period.

Their results show that the weekly mining power of a single entity has never exceeded 21% of the overall power. In contrast, the top Ethereum miner has never had less than 21% of the mining power. Moreover, the top four Bitcoin miners have more than 53% of the average mining power. On average, 61% of the weekly power was shared by only three Ethereum miners.

### 2.6.3 Deanonymization of users in the network

Any real-world entity that wants to exchange bitcoin needs to have at least one address. The address itself does not contain information about the identity of its owner, but an address can be linked to an identity in different ways. Usually, when an address is used in a transaction as input (or output), the receiver (or the sender) of the the sum knows the identity of the sender (or the receiver) or at least its affiliation. Hence, performing transactions with someone can deanonymize an address. This was done in [11], where the authors, in order to link addresses to people, performed the following activities: mining with pools, keeping wallets with different wallet services, trading with exchange services, purchasing goods from different vendors and with different payment systems, gambling online, and using mixing and laundry systems. Furthermore, it is common to find on the web people posting their address to receive donations or funds and these data have been collected in `walletexplorer.com/`.

An orthogonal but still interesting work is presented in [7] where a new methodology is presented to reveal IP addresses of Bitcoin users.

Chapter 3

# Analysis of the First Level in Coinbase Flows

In section 2.2 we discussed about the importance of bitcoin mining and in section 2.6.1 we saw the related threats. Nevertheless, we still have to understand what happened and what is now happening in terms of pools' behaviors, what's their mining power and how are they related to each other. We will start by analyzing the first level of coinbase flows, where miners receive the newly minted coins.

There are currently few websites[1] that show the hash rate percentage of each pool. These services provide pie charts and stack plots representing the evolution of mining and pools' shares, and despite some data is available, it is not explained how these results are obtained.

Our first contribution is to create a data set containing all the information needed for assigning blocks to miners, for studying the relationships between pools and to provide details about payout addresses and coinbase tags used by pools. This data set will be made public to help and foster the research in the field, but also to allow us to proceed with our analysis to deeper levels of coinbase flows in the next chapter.

## 3.1 Collecting Data

The data set includes three JSON files: the first contains information about mined blocks, the second about payout addresses used in coinbase transactions and the third about mining pools. We will describe their content and the process to create

[1] https://blockchain.info/pools, https://btc.com/stats/pool

them in the following sections. For this work, we consider all the blocks between the Genesis Block and block 514,700, mined on April, 6th 2018.

### 3.1.1 Sources

In the following paragraphs, we present all the possible data sources, namely the blockchain, the APIs offered by block explorer websites [5], information publicly available on GitHub [6], and the association between address and real-world entity performed in [10].

#### Coinbase Transactions from the Blockchain

By being a publicly available resource, the Bitcoin blockchain lends itself well to conduct independent studies. The AIT is running a Bitcoin node with the Bitcoin Core client, which makes it easy to query the blockchain and get back blocks and transactions in a JSON format. Starting from block 514,700 and going back to block 0, we create two python dictionaries, one for blocks and one for addresses. The procedure is presented in Algorithm 1. For each block we have the coinbase tag, the list of payout addresses and the timestamp, while for each payout address we have the list of mined blocks, the list of coinbase tags used for each block and the amount of BTC received through mining.

**Algorithm 1** Creating dictionaries for *blocks* and *addresses*.

```
define $addresses$ dictionary
define $blocks$ dictionary
for $block\_height = 0$ to 514700 do
    get coinbase transaction
    get $payout\_addresses$ and $BTC$ received
    get $timestamp$ of block
    get $coinbase\_tag$
    for $address$ in $payout\_addresses$ do
        increment $addresses[address][received\_BTC]$ by $BTC$
        append $block\_height$ to $addresses[address][mined\_blocks]$
        append $coinbase\_tag$ to $addresses[address][coinbase\_tags]$
    $blocks[block\_height][coinbase\_tag] \leftarrow coinbase\_tag$
    $blocks[block\_height][payout\_addresses] \leftarrow payout\_addresses$
    $blocks[block\_height][timestamp] \leftarrow timestamp$
```

To retrieve coinbase tags we decoded each coinbase input and looked for specific strings. In particular, we used the tags in [6] and we also used regular expressions to find strings like "pool", ".com", "BTC", "mine", "Bit", "coin" which allowed us to discover pools and, after some manual cleaning, to assign more blocks.

### Blocktrail API

Until April 16th, 2018 Blocktrail was providing an API to get details about blocks and transactions. For the purposes of this work, the interesting information available was the name of the miner of each block. It is not explained, though, how this attribution is made and, as with all the other sources that we have, it must be taken with a grain of salt, even more so considering the fact that Blocktrail was bought by Bitmain[2].

### Known Addresses

There are few other sources available on the web that have collected information about pools and addresses like in [6] and in `walletexplorer.com`. The first reports addresses known to be associated to mining pools and which coinbase tags they used. The second, instead, offers a broader knowledge about the entities behind addresses thanks to the multiple-input clustering heuristic and thanks to a set of relationships they found between addresses and real-world entities. This means that with this kind of sources we might be able to map addresses, and in particular payout addresses, to mining pools, crypto-currency exchanges, wallet services, websites and more. If a known address—an address for which the owner is known—is among the payout addresses of a coinbase transaction, we are able to attribute that block to the owner of that address.

## 3.1.2 Pools, Blocks and Addresses Files

The next step is to use all the information retrieved from the sources presented in the previous paragraphs to create our three JSON files. The first file to create contains details about each mining pool namely: payout addresses used, mined blocks, revenues from mining, coinbase tags used and any other alternative name of the pool.

The second file in our data set contains details about each mined block and, in addition to what has been introduced in section 3.1.1 about the *blocks* dictionary, we also include the block attributions using the Blocktrail API and the known addresses. When combining these three different inputs, we get four possible outcomes: (I) the three sources attribute the block to the same miner, (II) the three sources attribute the block to at least two different miners, (III) at least one source is able to attribute the block to a miner, (IV) none of the three sources is able to attribute the block to a miner. In the first and second case, we are able to assign the block to a miner; in the third case, we have a conflict and if we are not able to

[2] The API is now hosted by BTC.com.

solve it manually, we don't assign the block; in the last case, we don't assign the block.

The third file reports information about payout addresses coming from the python dictionary *addresses* obtained from Algorithm 1. Moreover, we include to each payout address its cluster ID and, after the block attribution procedure, the list of miners for which the address received bitcoin, which can be seen as its mining history.

## 3.2 Methodology

### 3.2.1 Computing the Mining Power Distribution

As described in section 3.1.1, a block can be assigned to a pool using different sources and combining all of them allowed us to get a smaller the number of unattributed blocks compared to cases where each single source was considered separately. To compute the mining power distribution, we count the number of blocks mined by a pool, or more generally a miner, within a block interval of $N_b$ blocks and each pool will get a percentage of the total hash rate. Minor pools or other miners for which the sum of the $N_i$ percentages is below the threshold of 1% will not be represented.

To see how mining power distribution evolved over time, we compute these percentages for $N_i$ intervals, each of $N_b = 2016$ days, as the difficulty is constant for 2016 blocks, roughly 2 weeks. The starting time of the first interval is chosen so that it corresponds to the time of a change in difficulty.

Using these percentages, we can measure the degree of decentralization in bitcoin mining with different concentration ratios, which indicate what is the market share of the $N$ largest firms in a market, so for example $CR_4 = 0.8$ means that the 4 largest firms own 80% of the market.

### 3.2.2 Mining History of Clusters

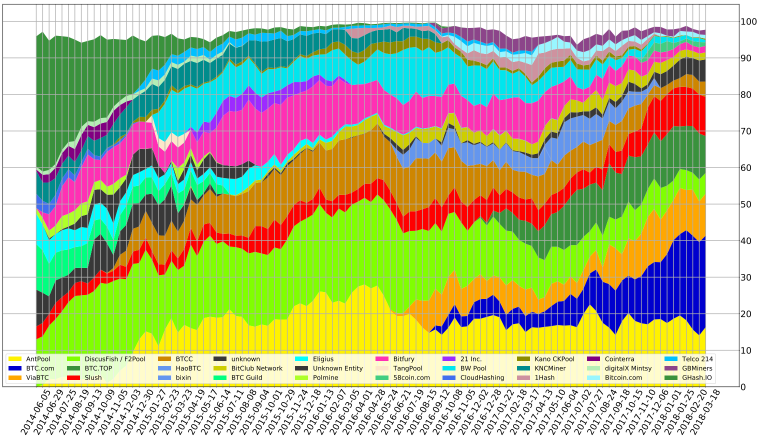
To see if and how mining pools are related to each other already at the first level of a coinbase flow, we analyze the payout addresses and the pools for which they received block rewards. In particular, for each payout address that we retriev in section 3.1.2, we get its cluster ID and the set of pools for which it mined and create a dictionary with cluster IDs as keys and sets of pools as values, which is basically joining the mining history of the addresses in the cluster and create a mining history for the cluster.

## 3.3 Results

### 3.3.1 Mining Power Distribution

Figure 3.1 shows how mining pools gained and lost mining power compared to the overall network over a 4-year period. For every two weeks ($N_b = 2016$), over a 4-year time span ($N_i = 105$), between 2014-03-18 and 2018-03-18, we compute the percentage of blocks mined by each pool. The white region represents minor pools or other miners for which the sum of all its $N_i$ percentages is below 1%.

Figure 3.1. Evolution of mining pools' hash rate percentages over a 4-year period. Each percentage is computed on a 2016-block interval ($\tilde{2}$ weeks). The *unknown* stack represents the share of unassigned blocks while the white stack represents the blocks assigned to other smaller miners.



It is easy to notice that some of the major pools in 2018 did not even exist in early 2016, e.g., BTC.com, ViaBTC and BTC.TOP. Vice versa, most of biggest pools in 2016 like BTCC, Bitfury and BW Pool are now smaller players. Also, the percentage of unassigned blocks has increased at the end of 2017.

One further remark is that since January 2018, AntPool, BTC.com and ViaBTC hold more than 50% of the overall mining power. As explained in section 2.4.1, we want to stress the fact that mining bitcoin is not so decentralized as usually claimed, rather the decision and the power of one, or maybe two companies can decide the future of the currency itself.

To further develop on the issue of decentralization, Figure 3.2 shows the trend of the decentralization indices and of the mining pools' shares during the same time period analyzed in Figure 3.1. For this plot, we do not show mining pools that never reached a share above 7%, despite their are of course used to compute the concentration ratios. It is clear that from mid-2017 mining centralization started to increase and from November 2017, the $CR_2$ corresponds to the Bitmain share (BTC.com and AntPool).

Despite this non-negligible concern, it is necessary to specify that Bitmain's pools are public and members are free to join or leave, so any action taken by the company that would harm the Bitcoin network, would also be detrimental to the interests of Bitmain itself as miners would move their hashing power away from its pools. What remains unclear is the percentage of hash power controlled by Bitmain within its pools and not owned by other entities. We will see more on this in section 4.1.2.

One possible, but controversial solution to this centralization of power is switching to a Proof-of-Work algorithm that is ASIC-resistant with a hard-fork[3], so that ASIC manufacturers will lose their dominant position and hash power will be provided by smaller and more decentralized miners around the world, as it was during the GPU era. Nevertheless, some critics say that hardware manufacturers will always be able to design specialized chips to make any general-purpose hardware obsolete and inefficient and we should not fear a 51% attack as developers and the 49% of honest miners can always hard-fork the chain and ignore the attacked chain.

### 3.3.2 Cross-Pool Mining with Payout Addresses

With the cluster's mining history introduced in section 3.2.2, it turns out that some clusters mined for different pools and this can happen mainly for two reasons. The first is that in the cluster there is at least one address linked to multiple pools, as in the case of cluster *259820950* which has one address[4] related to BTC.com and waterhole.io. The second is that there are two or more addresses with different pool tags, as in the case of cluster *304700830* with one address[5] associated to HaoBTC and another address[6] associated to BTC.com.

When looking for clusters with multiple pools, we got a long list. The majority of them was formed by a combination of the following pools: Eligius, P2Pool,

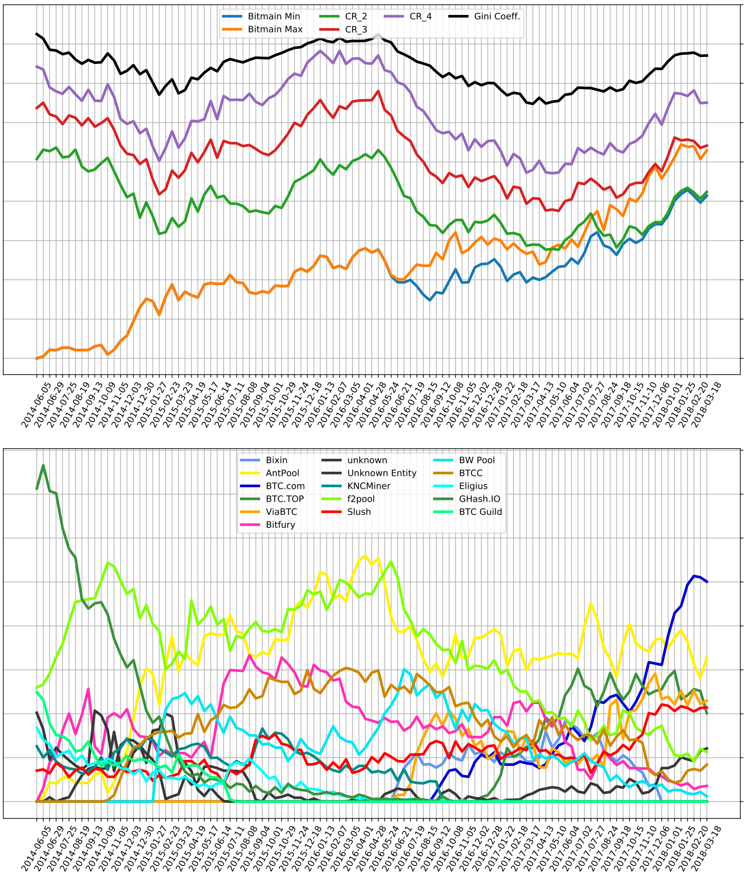
[3] A hard-fork happens when new consensus rules are applied only by a fraction of the network and the remaining part continues validating transactions using old rules. This leads to two different chains with two different protocols

[4] Full address: 1FLH1SoLv4U68yUERhDiWzrJn5TggMqkaZ.

[5] Full address: 1KsFhYKLs8qb1GHqrPxHoywNQpet2CtP9t.

[6] Full address: 1Csp5E15vii9HvPhLy3gkgWYXShXB8aQCw.

Figure 3.2. Mining shares of the major pools from Figure 3.1 and the corresponding centralization indices. $CR_N$ is the concentration ratio using the largest N pools, *Bitmain Min* represents the share of AntPool and BTC.com and *Bitmain Max* represents AntPool, BTC.com and ViaBTC.



KNCminer, HHTT and ckpool and this is a normal result because those pools used to make blocks with coinbase transactions with many output addresses and each address was owned by a single miner and not by the pool manager. Nevertheless, we also got other interesting and suspicious clusters that are reported in Table 3.1.

Table 3.1. Some of the clusters associated to multiple pools. The tags are obtained as explained in section 3.2.2.

| Entity | Cluster | Address | Pool | Blocks |
|---|---|---|---|---|
| Bixin | 304700830 | 1Csp5E15vii9HvPhLy3gkgWYXShXB8aQCw | BTC.com | 446292, ... |
| | | 1KsFhYKLs8qb1GHqrPxHoywNQpet2CtP9t | Bixin | 445401, ... |
| | | 13hQVEstgo4iPQZv9C7VELnLWF7UWtF4Q3 | Unknown | 497859, ... |
| BTC.com | 259820950 | 1FLH1SoLv4U68yUERhDiWzrJn5TggMqkaZ | BTC.com | 482221, ... |
| | | | Waterhole | 475040, ... |
| AntPool | 56139230 | 1Mp3atombUR6iLLYoLEdN9uPfPscsKBXfi | AntPool | 318734, ... |
| | | | P2Pool | 326655, ... |
| BTC.TOP | 31439813 | 147SwRQdpCfj5p8PnfsXV2SsVVpVcz3aPq | BTC.TOP | 474000, ... |
| | | | Canoe Pool | 500540, ... |
| ??? | 41016769 | 1PsQV5i36pUX12ucLViQmrfwyDQwyxym3W | BCPool | 310357 |
| | | 1JFGdjqtUfhbMcQxcMajo8Z71NFESJTUA7 | BCPool | 311559 |
| | | 12xQyJSnpYMsjH55ditWR3TWNEPYtDVJxT | BitAffNet | 312570, ... |
| | | 1BCESmGbSpDpEUbi6CLgc5DF27i6MRQMak | BitAffNet | 312664 |

Cluster *304700830* is formed by 7 coinbase output addresses, one related to Bixin, one to BTC.com, one to P2Pool, one to CKpool, one to Eligius, and two have no tag. The address[7] related to BTC.com has mined block 446292 and 446216. The address[8] related to Bixin has mined instead 2835 blocks (the first was 445401 and the last was 497846). The address related to Eligius mined only block 403662, together with other 135 addresses. Similar result for the addresses related to P2Pool and CKpool, that mined few blocks together with other addresses. One of the unknown addresses mined just one block alone (i.e., it was the only output address of the coinbase transaction), while the other address[9] with no tag mined 316 blocks (the first was 497859 and the last was 514645), always alone. Finally, according to walletexplorer.com these addresses are owned by HaoBTC, which was probably used as wallet service by the miners.

Cluster *259820950* is formed by just one coinbase output address[10] but it has signed two blocks (482,221 and 482,059) with the *BTC.com* tag and other 35 blocks

[7] Full address: 1Csp5E15vii9HvPhLy3gkgWYXShXB8aQCw.

[8] Full address: 1KsFhYKLs8qb1GHqrPxHoywNQpet2CtP9t.

[9] Full address: 13hQVEstgo4iPQZv9C7VELnLWF7UWtF4Q3.

[10] Full address: 1FLH1SoLv4U68yUERhDiWzrJn5TggMqkaZ.

(475040 the first and 467269 the last) with the *waterhole.io* tag. Some other blocks mined by this address have no tag at all. By looking at the waterhole.io website, one notices that Bitmain is indeed among the partners of the pool.

Cluster *56139230* is formed by 13 coinbase output addresses, all of them mining for AntPool, but one[11] of them mined 6 blocks for AntPool and 5 blocks with P2Pool (see Table 3.2). Similarly, cluster *23013447* has one coinbase output address[12] that was the payout address for AntPool for 334 blocks (between block 286,681 and 318,637), but it also was among the output addresses of coinbase transactions of blocks mined by P2Pool and the only output address of coinbase transactions of 28 unsigned blocks. What is the reason behind this fact? It is not clear yet, but certainly AntPool was not a big player in the summer of 2014 as can be seen from the stack plot in Figure 3.3.2.

Cluster *31439813* is formed by five coinbase output addresses, all of them mining with BTC.TOP, but one[13] of them, after mining for BTC.TOP (474000 the first mined block and 497286 the last), mined also 149 blocks (500540 the first and 514610 the last) signing them with the Canoe Pool tag.

Cluster *99788327* is unfortunately a so-called super-cluster where we have many addresses and not necessarily owned by the same entity.

Cluster *41016769* is formed by 4 coinbase output addresses, 2 of them mined blocks for bitcoinaffiliatenetwork and 2 for BCPool.IO and they where the only output addresses of the coinbase transactions, so they must belong to the pools.

Cluster *36175720* has 6 coinbase output addresses and one of them[14] mined blocks with Eligius till August 2015—when BTC.TOP was not active yet—and then became the payout address of BTC.TOP.

Cluster *34997292* and cluster *252222451* have both three addresses and they where used as single payout addresses by the pools in the clusters (Bitcoin Russia, Bitfury and digitalX Mintsy), except for P2Pool.

It is not clear yet how all these clusters are linked to different mining pools. In some cases, it might be a legitimate phenomenon, as many pools are open and the coinbase output addresses can be owned by many different single miners (e.g., Eligius or P2Pool) and the same address can change mining pool, but in some other cases, where there is only one coinbase output address, usually owned by a mining pool, we can suspect that there is some sort of relationship between the pools appearing in the same cluster.

[11] Full address: 1Mp3atombUR6iLLYoLEdN9uPfPscsKBXfi.

[12] Full address: 1GuMujABuc8kvzDTyVJFpcf4vszPUgsjiU.

[13] Full address: 147SwRQdpCfj5p8PnfsXV2SsVVpVcz3aPq.

[14] Full address: 1Eys1tWf2HDwbBctJ8KG4STFYQHH9eV8L1.

Table 3.2. Blocks mined by address 1Mp3atombUR6iLLYoLEdN9uPfPscsKBXfi and the associated pools. The coinbase transactions associated to AntPool blocks have only one output address, while the ones associated to P2Pool have around 340 output addresses.

| Block Height | Pool |
|---|---|
| 318734 | AntPool |
| 326655 | P2Pool.org |
| 326763 | AntPool |
| 326804 | P2Pool.org |
| 326869 | AntPool |
| 326875 | P2Pool.org |
| 327019 | AntPool |
| 327271 | P2Pool.org |
| 327293 | AntPool |
| 327315 | P2Pool.org |
| 327326 | AntPool |

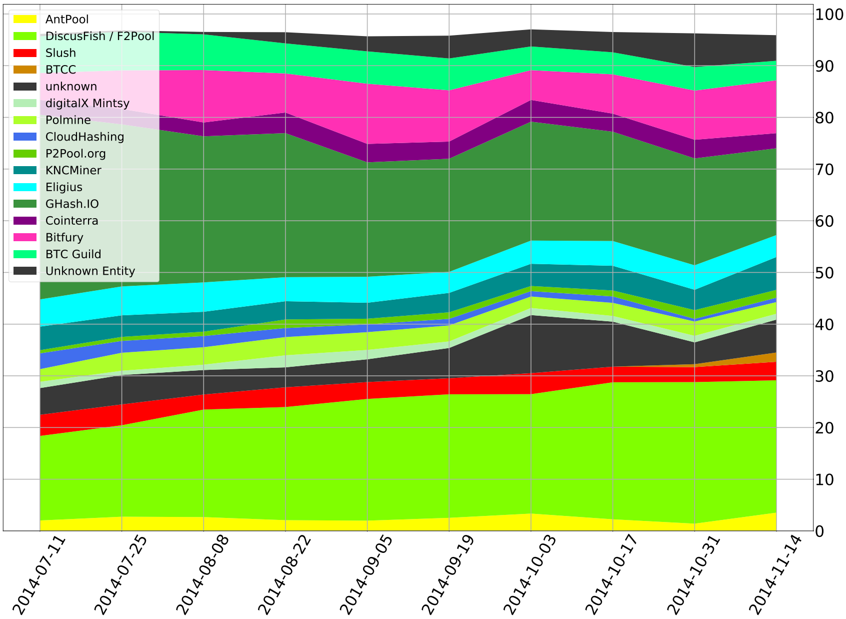


Figure 3.3. Stack plot of the hash rate distribution between 2014-11-14 and 2014-06-27, which corresponds to the period of the blocks in Table 3.2. The percentages are in accordance with the heat map reported in [1].

# Chapter 4

# Analysis of Deeper levels in Coinbase Flows

The second contribution of this work sheds some light on what is happening at deeper levels of coinbase flows, namely who are the pool members, what's their behaviour, what are their relationships with pools and with the results from the analysis of the first level. To do this, we will need to retrieve more data from the Bitcoin blockchain and also to rely on the multiple-input clustering heuristic for revealing identities associated to addresses.

To achieve these goals, there are at least two possible ways to proceed, both with some pros and cons. Our first approach is to retrieve coinbase flows for blocks assigned to the major pools (i.e., BTC.com, AntPool, ViaBTC, BTC.TOP, Slush, F2Pool), starting from the coinbase transaction and moving to deeper levels where payout transactions appear, and try to develop a model to describe their behaviour. The second method instead acts in the opposite direction in that it first finds a payout transaction of a pool where some coins are given to pool members and then it goes back to the coinbase transaction where those coins were minted.

While the analysis of the first level started from the Genesis Block, here we focus our attention to a period of time that goes from block 300,000 to block 514,700 as most of the current major pools were not active before this period.

## 4.1 From Coinbase Transactions to Payout Transactions

The idea behind this approach is that we can track which address received which sum of BTC and at which level of the coinbase flow and what role it plays in

Bitcoin mining.

### 4.1.1 Retrieving the Data

The challenge here is that the blockchain is a singly linked list and it is trivial to get the transaction where an input comes from, but doing the opposite, i.e., seeing where an output has been spent, is not. Thanks to the importance of and the interest for this information, there is an interface provided by `blockchain.info` that allows us to see and get the JSON format of how outputs have been spent. By recursively exploring each output of a transaction, we can build the tree of a coinbase flow.

The problem that arises with this procedure is to understand at which depth one should stop growing the tree. Our first decision is to stop at a level where we find an address owned by a crypto-currency exchange or when we have at least one hundred child nodes. The motivations behind these two choices are that in the first case, if we go one step deeper, we would look at coins coming from activities unrelated to bitcoin mining, while the second case is a threshold that we set in case we don't find any exchange in the coinbase flow. We understand this threshold is not backed by any scientific reason, rather it is just a lower bound estimate for the number of members in a pool.

The first version of the data set for this analysis consists of 214,700 JSON files where each file represents a coinbase flow. To this raw data we need to include two more pieces of information. Each address must be associated to its cluster, represented by an ID number, and the corresponding real-world entity, represented by a cluster tag obtained as explained in the next section.

#### Mapping Clusters to Entities

In our analysis, a key role is played by the ability to link a cluster of addresses to a real-world entity and assign a representative string or tag to it.

To associate cluster ID to a tag we proceed as follows: if the cluster contains at least one address in [6], its tag is used for the cluster, otherwise the tags provided by `walletexplorer.com` are used. If none of the two previous alternatives can provide an identity, we rely on the mining history of the cluster as explained in section 3.2.2, where the tag is made of a set of pool names. This mapping, of course, does not always allow us to reveal the identity of a cluster, for example when we have a cluster associated to many pools like Eligius, P2Pool and so on, but in cases where we have one pool using a single payout address like BTC.com, it is a useful information to have. In order to distinguish this source from the others we add to the set of pools the string "cb" to indicate that tags come from the coinbase analysis.

We decided to use [6] as first choice because it is the most reliable source of links between entities and addresses, while the cluster mining history, despite being a solid mapping between addresses and pools, might include less tags than walletexplorer.com.

```
{'13DQo..._249123673_None':
 [{'from': '18cBE..._220713833_ViaBTC',
   'received_BTC': 9.29387346,
   'redeemed_tx': 241145400,
   'when': 1490805010}],
'141tp..._249123673_None':
 [{'from': '18cBE..._220713833_ViaBTC',
   'received_BTC': 10.0999,
   'redeemed_tx': 240266114,
   'when': 1490805010}],
'1NYtB..._249123673_None':
 [{'from': '18cBE..._220713833_ViaBTC',
   'received_BTC': 10.0,
   'redeemed_tx': 241145400,
   'when': 1490805010}]}
```

Listing 1: Example of three nodes in one level of a coinbase flow. Each node is a dictionary with a key made of: *address*, *cluster ID* and *cluster tag* and its value is a list of dictionaries where *from* is the sender of *received_BTC* with timestamp *when*. *redeemed_tx* is the pointer to the transaction where the node spent the received sum.

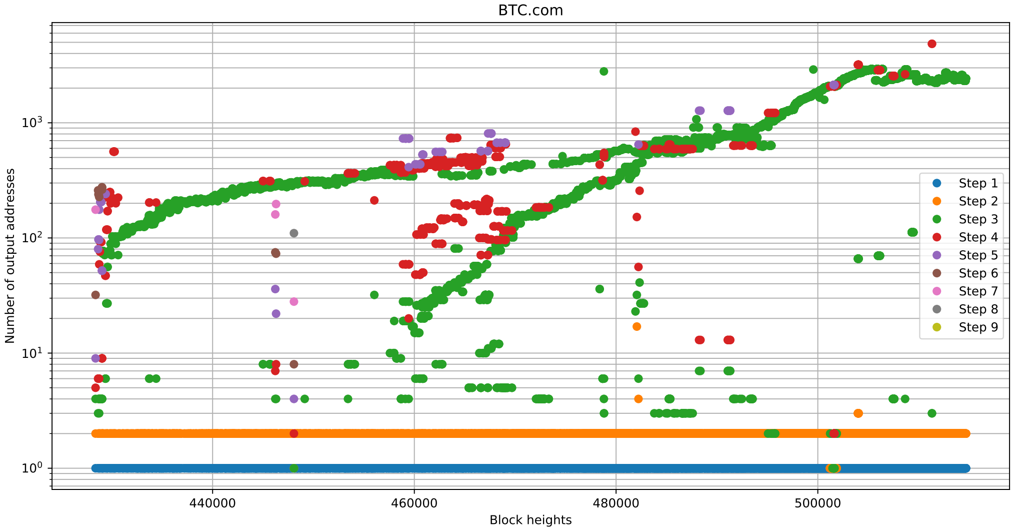
### 4.1.2 Payout Patterns and Pool Trends

With the collected data we can immediately see how many addresses we have at each step of a coinbase flow and get a sense of how and when a mining pool performs a payout transaction to split the block reward among its members. For some coinbase flows, it is pretty easy to find patterns in payout transactions, recognize the *paying address* and the pool members, while for others it is not and we will discuss some examples in the next sections.

#### BTC.com Payout Patterns

As can be easily noticed by looking at Figure 4.1, BTC.com is doing payout transactions in most cases at step 3, where we have many output addresses. If we

Figure 4.1. Payout patterns for BTC.com by looking at how many addresses are there at each level of a coinbase flow for all blocks mined by the pool.



analyze, as an example, the coinbase flow of block 509,997 sketched in Figure 4.2, we see how the structure matches Figure 4.1 and how quickly the number of total outputs grows at each step. In this example, the address performing the payout address, the *paying address*, i.e., *A2*, does not change.

In other coinbase flows, e.g., 463,049 or 463,273, the change address in a payout transaction differs from the input address and, in some other cases, it is not possible to identify it by looking at the sum of BTC received as change, which is usually a large sum. Furthermore, it may happen that the change address of a payout transaction does not continue the payments to individual miners, rather the coins go somewhere else. This is why we think trying to get all the possible payout transactions is inconclusive and not the best way to proceed.

As we will also see for all the other main pools, the payout address for BTC.com stays the same, usually for some months or more, and gets many block rewards as shown in Figure 4.2.

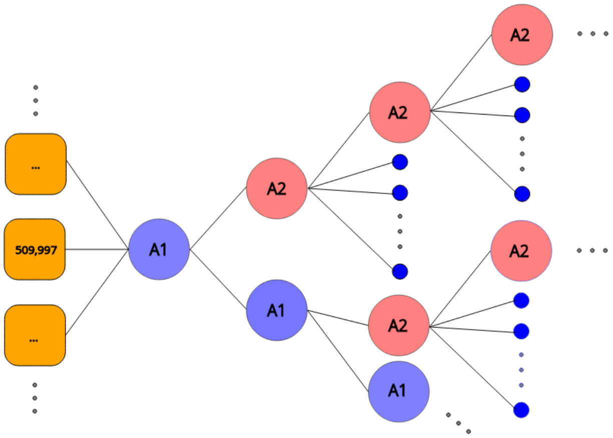


Figure 4.2. BTC.com coinbase flow example. The opaque-blue node is the payout address that gets block rewards, opaque-red nodes perform payout transactions to pool members that represented with smaller blue nodes. Rounded squares represent coinbase inputs of blocks mined by the pool. The size of the nodes tries to represent the amount of BTC received in a transactions.

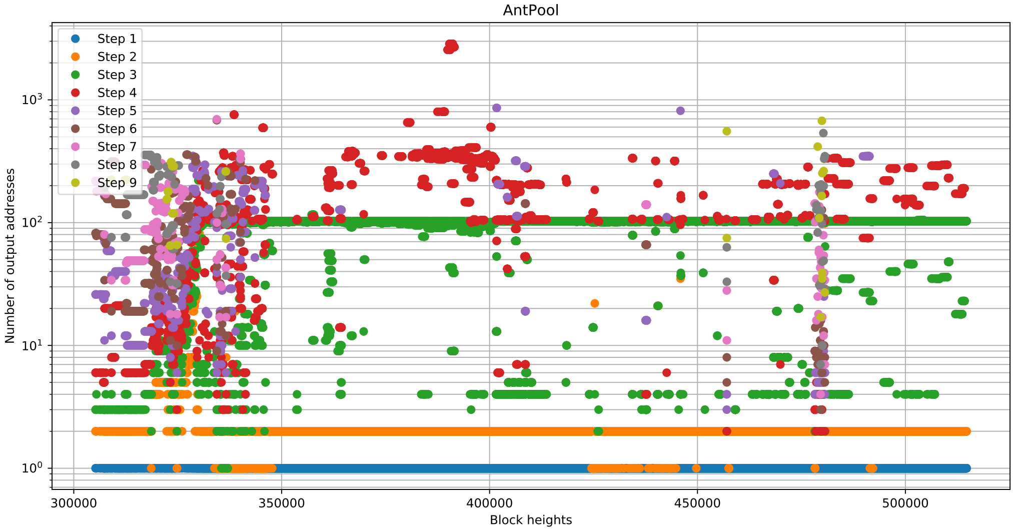
### AntPool Payout Patterns

The structure presented in Figure 4.2 is basically the same in most of the coinbase flows of blocks mined by AntPool. The main difference here is that the number of outputs in a payout transaction, as can be seen from Figure 4.3, is always around 100. The other difference is that the change address of a payout transaction is always changing and it is not necessarily the address receiving the largest sum from the previous payout transaction.

### ViaBTC Payout Patterns

If we have a look at Figure 4.4, we see the different patterns ViaBTC was and still is following for splitting the block rewards. Initially, a payout address was directly performing payout transactions and pool members were at step 2 and 3, as depicted in Figure 4.5. This lasted till before block 460,000 and then the payout address started transferring block rewards to other addresses in charge of doing payout transactions. At the beginning of this new pattern, there was always only one address at step 2, but more recently the pool decided to split these block rewards into equal parts of 10 BTC among few dozens of addresses and then, some of them would pay to pool members, that we see at step 3. These paying addresses,

Figure 4.3. Payout patterns for AntPool by looking at how many addresses are there at each level of a coinbase flow for all blocks mined by the pool.

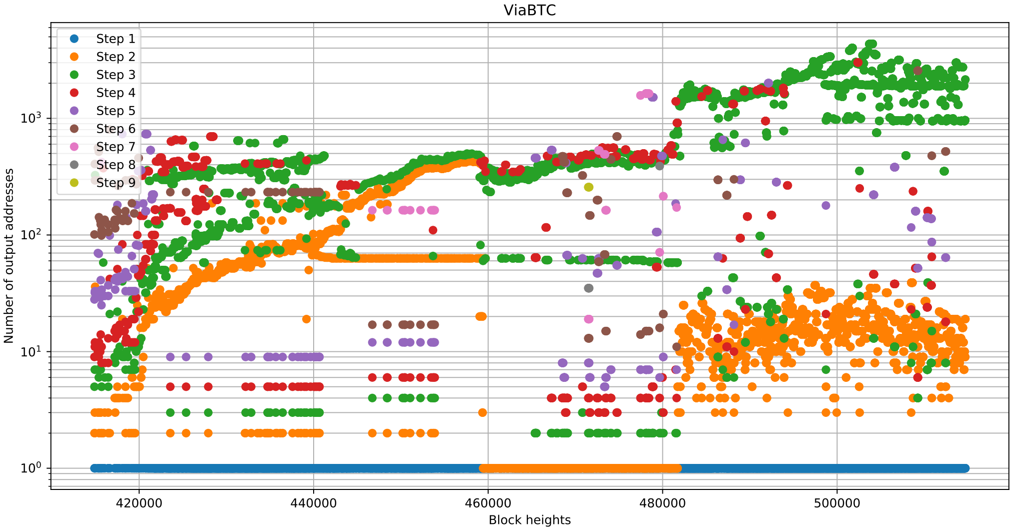


though, are always different. An example for this payout scheme is provided in Figure 4.6.

### Slush Payout Patterns

The payout behavior of Slush, the oldest pool, is much more chaotic as Figure 4.7 shows. Coinbase transactions in blocks mined by Slush have always had one single output, which is then sometimes used to pay pool members. By looking at the plot, we don't see any fixed number of output addresses representing pool members, as it was for BTC.com or AntPool, nor do we see a clear distinction of who is performing the payout transaction. By manually checking some flows, we saw that the main pattern for Slush is the one already presented in Figure 4.5, even though there are many other cases where there is an intermediate transaction before pool members receive their share. The general conclusion is that we have payout transactions at step 2 and 3 and their number of outputs varies between few tens to few hundreds of addresses.

Figure 4.4. Payout patterns for ViaBTC by looking at how many addresses are there at each level of a coinbase flow for all blocks mined by the pool.



### BTC.TOP Payout Patterns

Similarly to what happens with BTC.com, in Figure 4.8 we have a pretty clear pattern for payout transactions. We see that the number of pool members in a payout transaction increase with time, despite in the past we had many intermediate transactions before they got their rewards. Also the scheme used by BTC.TOP resembles the one in Figure 4.2, but in payout transactions the change address is not always the same as the input address.

### F2Pool Payout Patterns

More regular is instead the behavior of F2Pool, or at least apparently from Figure 4.9, where there is always one payout address, few addresses at step 2 and then we clearly have pool members. By manually analyzing some coinbase flows, we can draw the main pattern as simplified in Figure 4.10.

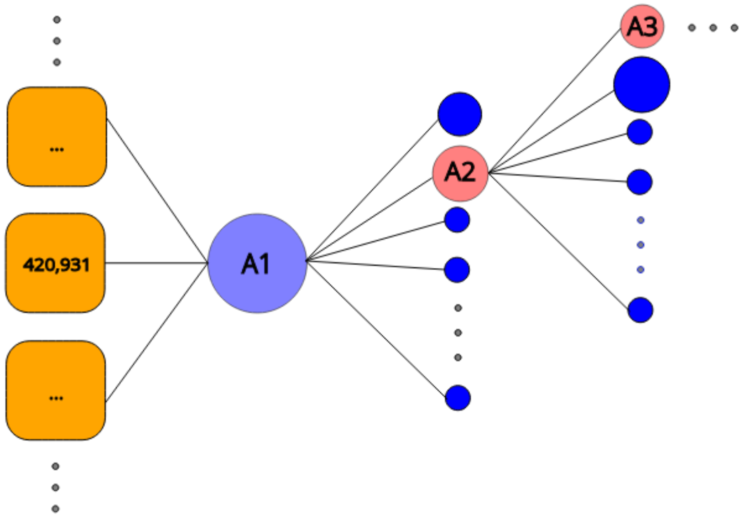


Figure 4.5. ViaBTC coinbase flow example. The opaque-blue node is the payout address that gets block rewards, opaque-red nodes perform payout transactions to pool members that represented with smaller blue nodes. Rounded squares represent coinbase inputs of blocks mined by the pool. The size of the nodes tries to represent the amount of BTC received in a transactions.

### Clustering Addresses in Coinbase Flows

So far, we have seen how many addresses were involved in a coinbase flow, but numbers decrease if we consider clusters instead. Indeed, in Figures 4.11, 4.12, 4.13, 4.9 we see that many addresses are grouped together into clusters and in 4.1.4, we will analyze some of the known entities behind these clusters.

### Preliminary Discussion

After this detailed description of payout patterns for the main pools, there are some conclusions we can draw. The first one is that it is not trivial to retrieve and provide statistics about all the payout transactions and pool members, given all the possible variables to consider, i.e., number of output addresses, amount of BTC received, change addresses. The second one is that all major pools usually perform payout transactions within few steps from the coinbase, even though there are some block rewards that are not distributed among pool members at all, rather they are either still unspent or sent to other unknown addresses, likely unrelated to mining. The third conclusion is that when block rewards are split among pool

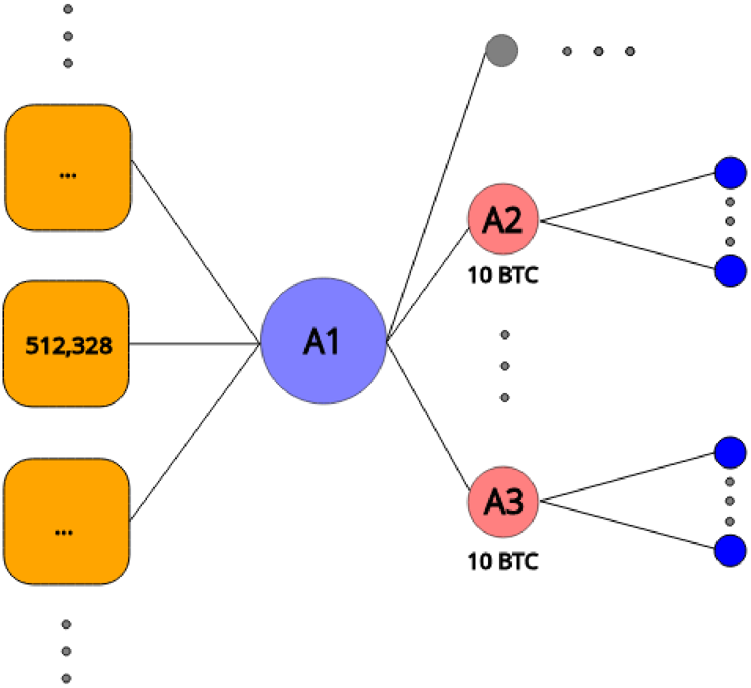


Figure 4.6. ViaBTC coinbase flow example. The opaque-blue node is the payout address that gets block rewards, opaque-red nodes perform payout transactions to pool members that represented with smaller blue nodes. Gray nodes are change addresses and do not follow any pattern. Rounded squares represent coinbase inputs of blocks mined by the pool. The size of the nodes tries to represent the amount of BTC received in a transactions.

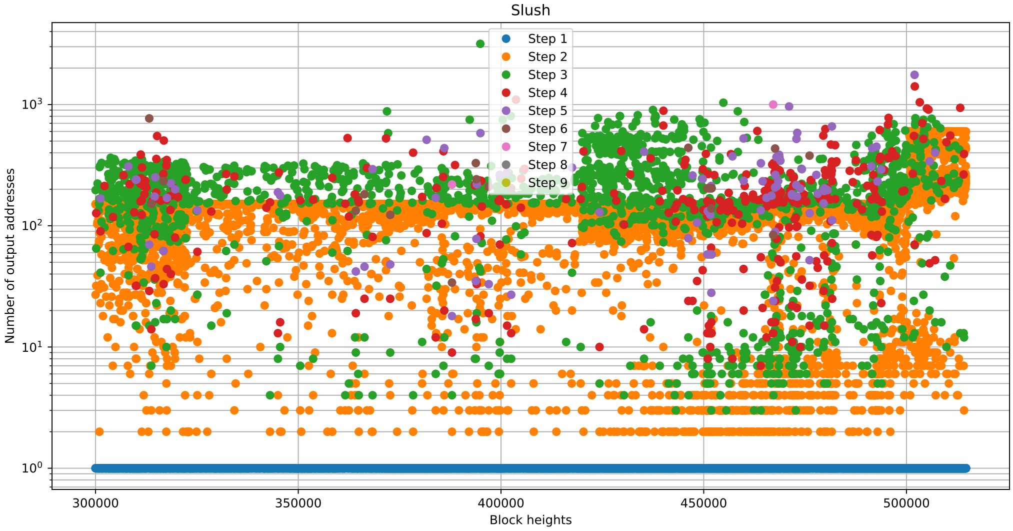
members, we basically always have a payout transaction in our data coinbase flow, either at step 2 or 3, if we stop retrieving data at a level with at least 100 outputs. The big limitation of this method is, of course, that we see only a small fraction of what is happening behind a mining pool.

### 4.1.3 Methodology

As stated at the beginning of this chapter, we want to characterize the behavior of pool members, reveal, when possible, their identity and understand their relationships with other pools. To address most of these aspects, we need to extract the information we are looking for.

First, we create a data structure where we save, for each pool, all the clusters

Figure 4.7. Payout patterns for Slush by looking at how many addresses are there at each level of a coinbase flow for all blocks mined by the pool.

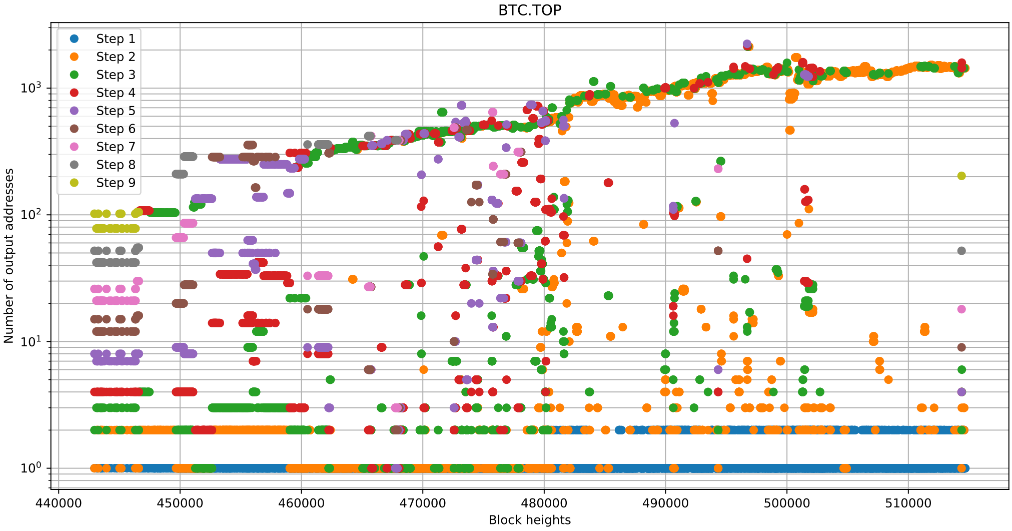


involved in the coinbase flows of the blocks mined by that pool and, for each cluster, we also save at which block height they appear. Relying on the multiple-input clustering heuristic, we then try to associate each cluster to its real-world entity.

One should keep in mind that this procedure has some limits. Firstly, it does not consider pool members only, rather it includes all addresses somehow related to the distribution of block rewards and, from what described to far, distinguish between pool members and the other actors is not straightforward. Furthermore, the biggest drawback here is that we are missing a lot of data about pool members, as explained in section 4.1.2, where payout transactions are performed in a chain-like fashion. It is clear that any result following from this analysis, should be interpreted as a lower bound.

With this considerations in mind, we also look at the economic aspect of cross-pool mining. This means that instead of storing just the block height, we also add the amount of BTC received by each cluster. It must be specified, though, that here we consider only the sums received by the leaves of our trees, i.e., the coinbase flows. In this way, we avoid counting coins multiple times, for example if address $A_1$ receives 100 BTC from a payout address $A_0$ and then it splits the sum among

Figure 4.8. Payout patterns for BTC.TOP by looking at how many addresses are there at each level of a coinbase flow for all blocks mined by the pool.



pool members, assuming they are more than 100, we ignore the sums previously owned by $A_0$ and $A_1$.

### 4.1.4 Results

Keeping in mind that our data set does not contain pool members only, because our coinbase flows include also non-payout transactions, we observe 201965 different clusters[1] and more than 200 different tags related to a wide range of entities from exchanges and wallet services to gaming and gambling websites. Given the very nature of Bitcoin, though, most of the clusters, and so their addresses, do not have a real-world identity.

Not surprisingly, we find some clusters involved in cross-pool mining, but for some of them we are able to disclose their identity, providing interesting insights. There are at least two possible ways to sort these clusters: one is to use the amount of BTC received and the other is the number of pools in which the cluster appears.

[1] If one address does not belong to a cluster, it is considered a cluster itself.

Figure 4.9. Payout patterns for F2Pool by looking at how many addresses are there at each level of a coinbase flow for all blocks mined by the pool.

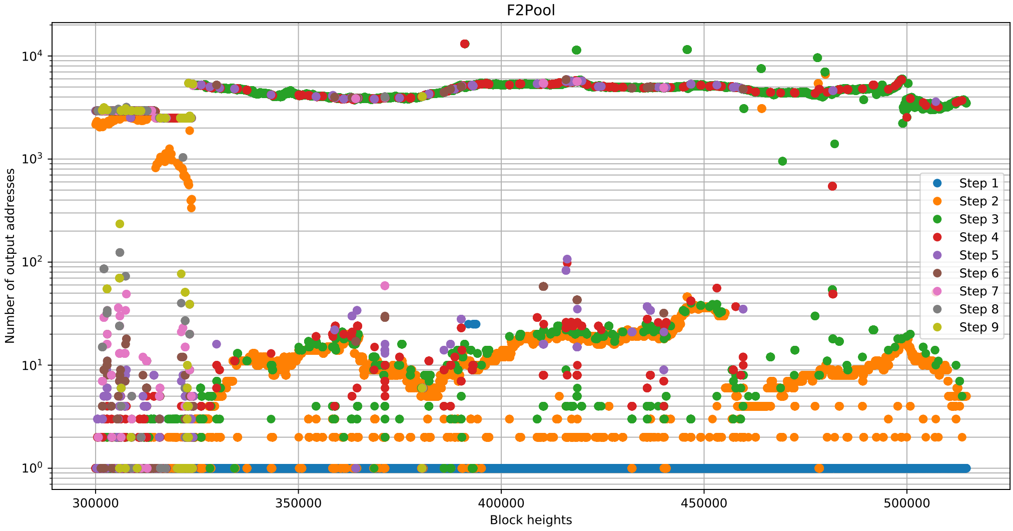


Table 4.1 contains the amount of BTC received by a cluster, identified by its tag, for mining with the pool in the column. We only report clusters with a tag that are active in at least two pools and receiving at least 500 BTC in total from cross-pool mining. If we remove the 500 BTC threshold and consider also unknown clusters, we get 641 entities involved in cross-pool mining.

Perhaps, even more interesting is to see how much and when clusters received bitcoin from cross-pool mining. In what follows, we will discuss some cases where clusters are associated to mining pools, highlighting problems and limitations.

### Cross-Pool Mining of BTCChina

In our results, BTCChina, also referred to as BTCC, is behind three clusters involved in cross-pool mining and Figure 4.15 shows the amount of BTC received by each cluster from each pool. It is clearly associated to Slush, BTC.TOP, F2Pool and BTC.com, AntPool and ViaBTC, especially when we see dots of the same color forming a line, indication of how the hash rate of the cluster evolved compared to the pool's hash rate. One further fact that we discovered is that the addresses used by BTCC to receive shares of block rewards reappear across payout transactions

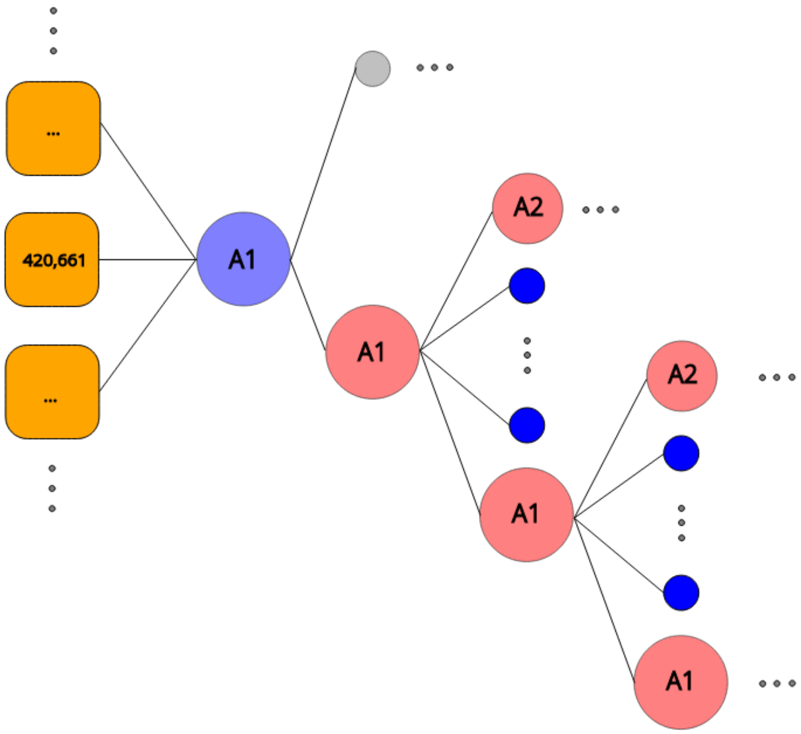


Figure 4.10. F2Pool coinbase flow example. The opaque-blue node is the payout address that gets block rewards, opaque-red nodes perform payout transactions to pool members that represented with smaller blue nodes. Gray nodes are change addresses and do not follow any pattern. Rounded squares represent coinbase inputs of blocks mined by the pool. The size of the nodes tries to represent the amount of BTC received in a transactions.

of a pool and even among outputs of a single payout transaction there are multiple addresses linked to BTCC.

A careful reader, though, could be surprised to see data points in the range of 50 or 100 BTC in Figure 4.15 as this is usually not a reasonable sum to receive for a pool member in a payout transaction. The cause of these values is that output addresses in a payout transaction usually accumulate a large sum of bitcoin to then send the rewards to a single address, owned by BTCC. An example is address

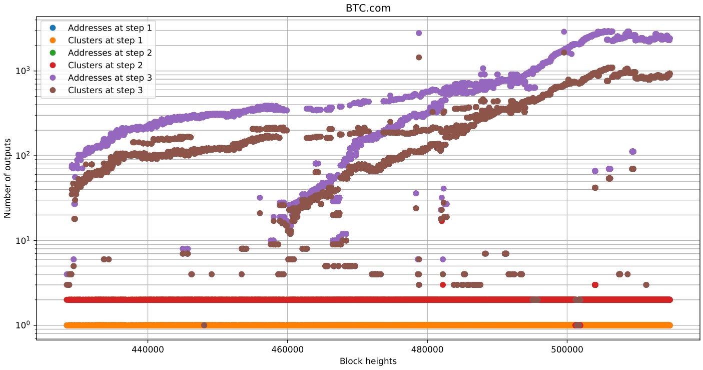


Figure 4.11. Comparison between number of addresses and number of clusters for the first three steps of coinbase flows in BTC.com.

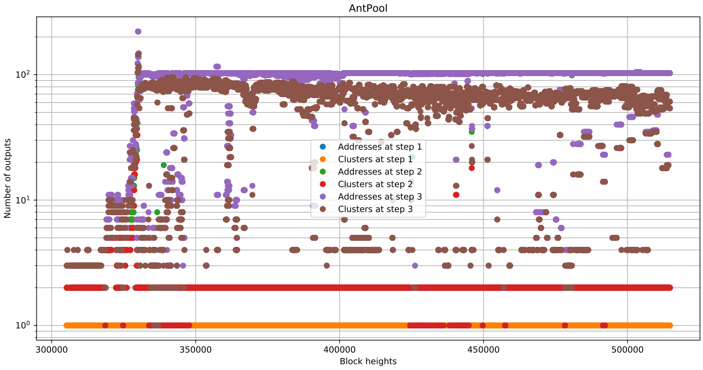


Figure 4.12. Comparison between number of addresses and number of clusters for the first three steps of coinbase flows in AntPool.

18e4w[2] which is collecting rewards from mining with AntPool and sending large sums to an address[3] related to BTCC.

One final remark on BTCC is that its cross-pool mining activity was performed

[2] Full address: 18e4wGHTsQCde4GTXjtpAJ297WRQpVAYrm.

[3] Full address: 1KuTTULMTVPFDzmidsb12MSe8Sv9pUDpJk.

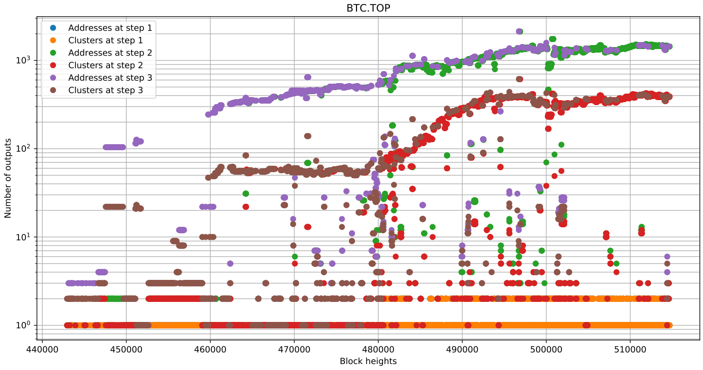


Figure 4.13. Comparison between number of addresses and number of clusters for the first three steps of coinbase flows in BTC.TOP.

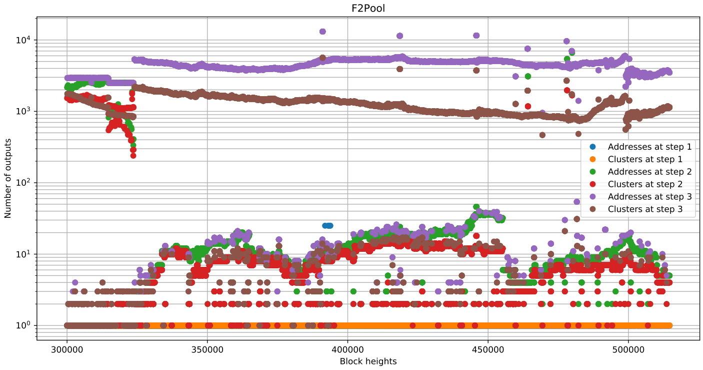


Figure 4.14. Comparison between number of addresses and number of clusters for the first three steps of coinbase flows in F2Pool.

while the pool was also active as a mining pool itself and, as we previously saw in Figure 3.1, its share of total hash rate in the network was decreasing. BTCChina is now a platform offering exchange, mining and wallet services and is owned by a Hong Kong-based blockchain investment fund.

### Cross-Pool Mining of BTC.TOP

BTC.TOP has been using a payout address[4] in coinbase transactions for quite a long time now, so it must be owned by the pool itself. With our analysis, we discovered that this same address was also active in 2016 as a pool member in F2Pool, receiving shares of rewards of blocks mined between block 416,697 and 423,305. Another address[5] belonging to this cluster also appeared among pool members of Slush and, in a less significant way, of AntPool and ViaBTC.

### Cross-Pool Mining of Bixin

Formerly known as HaoBTC, Bixin is not only a mining pool, but also a wallet and exchange service. This is probably why in Figure 4.17 we see that its cluster receives bitcoin from many pools and with different orders of magnitude. In particular, BTC coming from Slush and AntPool are shares of block rewards as their lower value suggest. On the other hand, higher values coming from ViaBTC, F2Pool and BTC.TOP are related to the wallet service that Bixin provides and should not considered as a reward from cross-pool mining.

### Cross-Pool Mining of Super-Clusters

In Table 4.1, a cluster linked to F2Pool is ranked as first, but some clarifications are needed. The main contribution to the total sum comes from the column of the pool itself because change addresses in its payout transactions are owned by the pool as shown in 4.10, so this is not an worrying behaviour.

The ambiguity in this cluster is that it contains addresses known to be owned by F2Pool[6], by Polmine[7] (mining pool from Poland) as listed in [6] and many other entities. Indeed, walletexplorer.com puts these addresses in a so-called super-cluster associated to MtGox and others. Consequently, we believe this should not be considered as a real cross-pool mining case.

A similar reasoning can be done with the cluster associated to Telco 214, an old mining pool. This cluster is formed by addresses known to be owned by the pool[8], but also by addresses related to the trading platform LocalBitcoins[9].

[4] Full address: 1Hz96kJKF2HLPGY15JWLB5m9qGNxvt8tHJ

[5] Full address: 1tkjtG7ZkhhxHYNgmvxaWxMrvLxdyQBi3

[6] Full address: 1KFHE7w8BhaENAswwryaoccDb6qcT6DbYY

[7] Full address: 1Nsvmnv8VcTMD643xMYAo35Aco3XA5YPpe

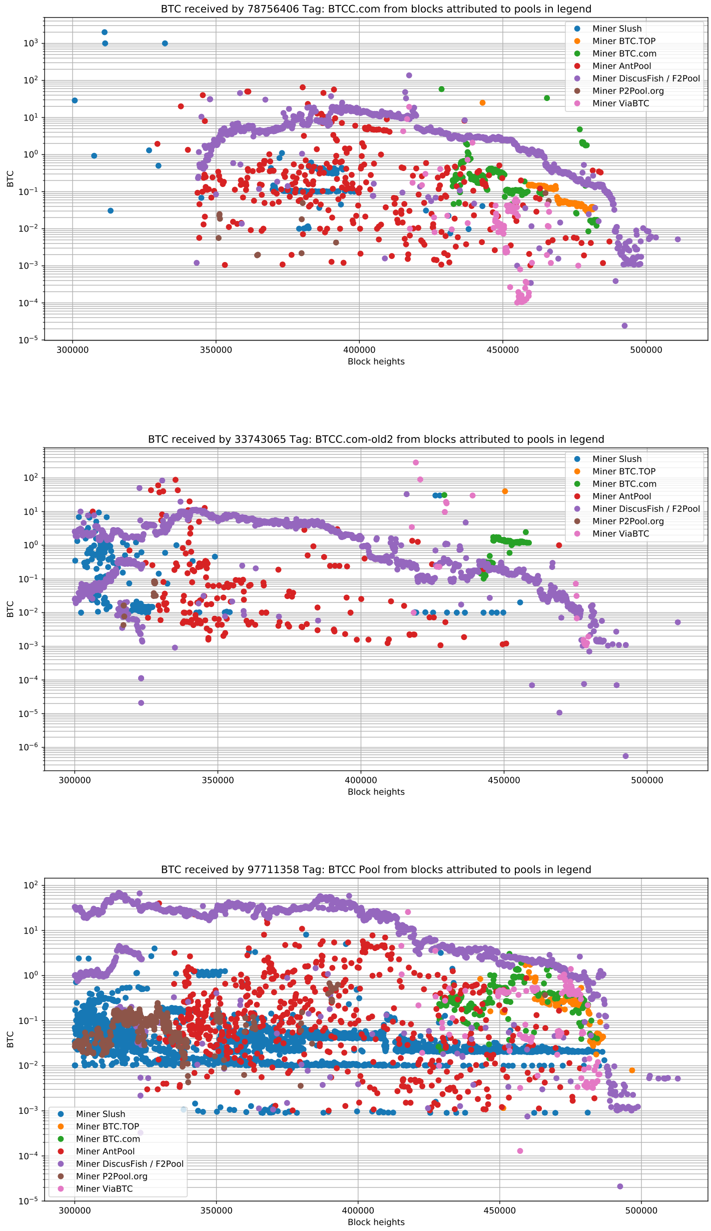
[8] Example: 1P4B6rx1js8TaEDXvZvtrkiEb9XrJgMQ19.

[9] Example: 1Lxfs8mPPEH8L4SgQ2AFmdKmdKnYNMveb7.

Table 4.1. Clusters involved in cross-pool mining, sorted by total BTC received. Each value in a cell represents the amount of BTC that the cluster in the row received from the pool in the column. Numbers are obtained as explained in section 4.1.3. Only clusters with a tag are shown.

| Slush | BTC.TOP | BTC.com | AntPool | F2Pool | ViaBTC | TOTAL | CLUSTER TAG |
|---|---|---|---|---|---|---|---|
| 2137 | 71 | 0 | 1001 | 2498350 | 29 | 2501663 | F2Pool |
| 2211 | 57136 | 8840 | 78731 | 211183 | 27963 | 386126 | Huobi.com-2 |
| 20588 | 16717 | 21004 | 49574 | 3567 | 32217 | 144429 | Bitstamp.net |
| 75747 | 20915 | 2879 | 9294 | 10238 | 15349 | 144065 | Poloniex.com |
| 28 | 133453 | 0 | 447 | 120 | 1 | 134050 | BTC.TOP |
| 1833 | 376 | 4699 | 35320 | 65265 | 8948 | 116484 | OKCoin.com |
| 410 | 34264 | 13337 | 1733 | 48229 | 13980 | 111991 | Bixin |
| 13499 | 11010 | 3957 | 18236 | 1566 | 10943 | 60214 | Bittrex.com |
| 456 | 662 | 5254 | 17394 | 2726 | 10943 | 37437 | OKCoin.com-2 |
| 6517 | 0 | 0 | 24418 | 1944 | 0 | 32881 | Bitfinex.com-old2 |
| 11118 | 32 | 3105 | 11931 | 672 | 3265 | 30139 | Xapo.com |
| 169 | 47 | 85 | 416 | 25627 | 77 | 26466 | BTCC Pool |
| 842 | 0 | 0 | 3732 | 18573 | 0 | 23189 | Huobi.com |
| 17208 | 0 | 0 | 47 | 99 | 0 | 17355 | MegaBigPower |
| 383 | 81 | 170 | 461 | 15773 | 456 | 17329 | BtcTrade.com |
| 10086 | 763 | 155 | 1921 | 2967 | 698 | 16701 | BTC-e.com |
| 3415 | 0 | 0 | 7599 | 3576 | 0 | 14591 | BTCC_cb |
| 2107 | 165 | 132 | 959 | 4414 | 3267 | 11170 | Telco 214 |
| 4054 | 33 | 152 | 621 | 5630 | 37 | 10530 | BTCC.com |
| 7688 | 303 | 630 | 364 | 124 | 226 | 9394 | Kraken.com |
| 161 | 1 | 5 | 4695 | 105 | 24 | 4995 | Hashnest.com |
| 162 | 40 | 132 | 430 | 3353 | 456 | 4577 | BTCC.com-old2 |
| 2306 | 81 | 105 | 133 | 379 | 131 | 3142 | Luno.com |
| 1 | 451 | 141 | 8 | 1735 | 99 | 2439 | phash.io_cb |
| 37 | 139 | 199 | 192 | 1744 | 20 | 2336 | Bter.com |
| 1187 | 1 | 5 | 68 | 110 | 40 | 1431 | Bitcoin.de |
| 959 | 0 | 0 | 156 | 61 | 0 | 1177 | BitPay.com-old |
| 0 | 0 | 0 | 1100 | 2 | 60 | 1162 | BW Pool_cb |
| 1028 | 0 | 0 | 1 | 4 | 0 | 1034 | CoinApult.com |
| 2 | 0 | 0 | 108 | 891 | 1 | 1004 | OkLink.com |
| 387 | 0 | 484 | 11 | 3 | 33 | 995 | Cryptopay.me |
| 317 | 5 | 9 | 4 | 395 | 137 | 873 | BitoEX.com |
| 114 | 267 | 13 | 123 | 46 | 246 | 816 | HitBtc.com |
| 277 | 24 | 3 | 102 | 74 | 169 | 655 | LocalBitcoins.com |

Figure 4.15. Cross-pool mining activity of BTCC.



BTC received by 78756406 Tag: BTCC.com from blocks attributed to pools in legend

BTC received by 33743065 Tag: BTCC.com-old2 from blocks attributed to pools in legend

BTC received by 97711358 Tag: BTCC Pool from blocks attributed to pools in legend

Figure 4.16. Cross-pool mining activity of BTC.TOP.

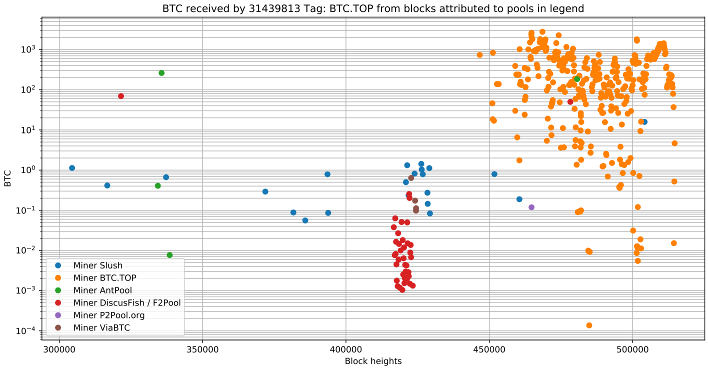


Figure 4.17. Cross-pool mining activity of Bixin.

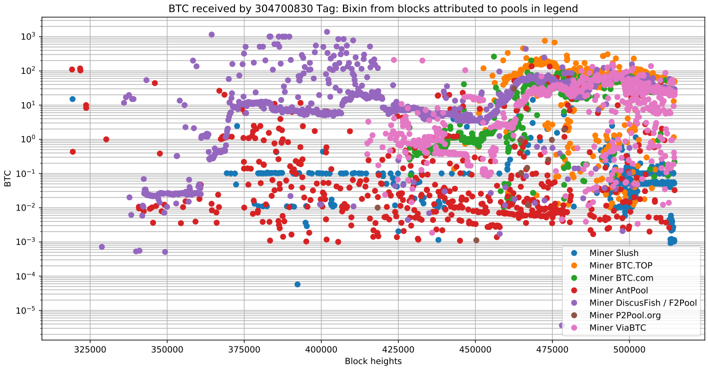
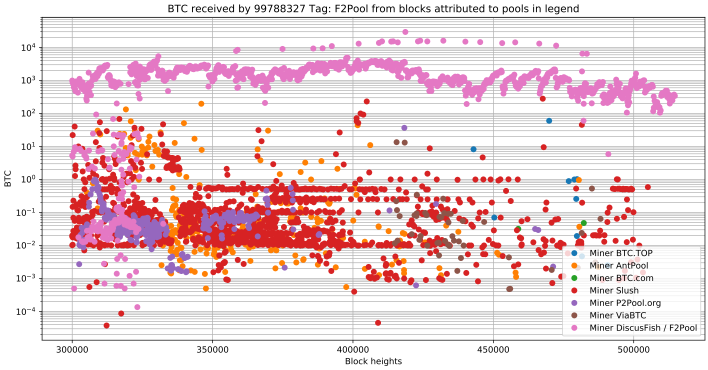


Figure 4.18. Cross-pool mining activity of a "super-cluster" with many tags. Some of them are: F2Pool, Polmine, MtGox, ePay.info.



# Chapter 5

# Pool Members and Pool Hash Rate Distribution

As explained in section 4.1.2, it is difficult to get all payout transactions performed by a pool due to different patterns it uses. Nevertheless, we saw that for some pools and for limited amount of time, it is worth trying.

## 5.1 BTC.com Miners

In Figure 4.1 and 4.2 we saw that BTC.com follows a specific pattern for payout transactions. In this section, we will try to collect all payout transactions in a time period that goes from 2018-02-19 23:40:54 when block 510,000 was mined to one day after block 514,032 was mined, 2018-03-19 03:51:48, which is roughly a month of mining activity (4032 blocks plus one day for considering payments to pool members for blocks mined during the last day).

### 5.1.1 Retrieving Payout Transactions

After our analysis in section 4.1.2, we think BTC.com used for a limited period of time always the same address[1] for payout transactions to pool members. This assumption might be wrong so we decided to compare the amount of BTC mined and the amount of BTC spent in payout transactions by this address. From the `blockchain.info` API, we collected all the transactions within the aforementioned

[1] Full address: 3EKc1EHHRVJrwBGFgvxd7ZJrKxjTHZZFhV.

period—1092 transactions in total— where this address is used as input[2] and saved for each output address the amount of BTC received.

### 5.1.2 Removing Pool Addresses

During this time frame, BTC.com received 13,058 BTC from mining while the sum of all the outputs in the collected payout transactions is 42,881 BTC. This number is much higher than the mining reward of the pool because we are also including the change address. If we ignore it, we get 13,714 BTC, still more than the mining reward. Among the output addresses, though, there is one address[3] that receives unusually big sums. This address is also receiving BTC from another address[4] owned by BTC.com through transactions without change addresses. It is not clear whether this is an address is controlled by BTC.com, but compared to the other output addresses it is definitely an outlier. By excluding it, the sum of BTC paid to pool members is 12,057 BTC, which is roughly 92% of the initial mining reward. In the following analysis, we decided not to consider these two addresses as pool members.

### 5.1.3 BTC.com Pool Members

Within our payout transactions we found 20,442 unique addresses, that can be grouped in 8,900 unique clusters with a total of 38 tags. Both for clusters and tags, we report the list, together with the percentage of BTC received and the number of addresses associated to each cluster or tag in Table 5.2 and 5.1.

Despite the large number of BTC going to unknown entities, from Table 5.1 we can estimate that at least 22.3% of BTC goes into exchange or wallet services. Furthermore, as already seen in section 4.1.4, there are clusters owned by other mining pools that receive a small share of block rewards and also old unknown miners still active.

Unfortunately, we cannot say whether the percentage owned by Bixin is related to its mining activity or to its other wallet and exchange services.

[2] It turns out that it is always the single input address.

[3] Full address: 18AR7ptjQxHfDrrGh1SKgv1jyFDNjJkaWw.

[4] Full address: 3L86WSsX94pirYw81zYjL8ii3hMRNRuYhQ.

Table 5.1. Tags related to members of BTC.com and their corresponding percentage of BTC received and the number of addresses. W: wallet, E: exchange, P:pool, M: old unknown miner, H: hash provider, F: faucet.

| Tag | % of BTC | No. addresses | Service |
|---|---|---|---|
| Unknown | 74.069385 | 13286 | ? |
| Bixin | 13.799274 | 1061 | W+E+P |
| Huobi.com-2 | 6.706924 | 964 | E |
| ckpool_ckpool.org_cb | 1.715818 | 985 | M |
| KNCMiner_unknown_ckpool_ckpool | 0.773891 | 1077 | M |
| Bittrex.com | 0.694330 | 348 | E |
| P2Pool.org_cb | 0.671016 | 931 | M |
| Poloniex.com | 0.353770 | 381 | E |
| Luno.com | 0.303509 | 258 | W+E |
| P2Pool.org_Eligius_cb | 0.246448 | 623 | M |
| Xapo.com | 0.223586 | 94 | W |
| Eligius_cb | 0.094405 | 79 | M |
| Bitstamp.net | 0.074131 | 57 | E |
| OKCoin.com-2 | 0.053856 | 5 | E |
| Cryptonator.com | 0.047675 | 80 | W+E |
| BitoEX.com | 0.042185 | 23 | W+E |
| CoinHako.com | 0.029796 | 4 | W+E |
| unknown_P2Pool.org_Eligius_cb | 0.020349 | 51 | M |
| Bit-x.com | 0.019484 | 5 | W+E |
| Bitcoin.de | 0.015391 | 26 | E |
| ckpool_ckpool.org_P2Pool.org_c | 0.011854 | 22 | M |
| Cryptopay.me | 0.008453 | 9 | W+E |
| CoinJar.com | 0.007911 | 11 | E |
| HitBtc.com | 0.006590 | 17 | E |
| GBMiners_cb | 0.001999 | 2 | P |
| ckpool_ckpool.org_Solo | 0.001871 | 12 | M |
| BitClub_Network | 0.001120 | 1 | P |
| Hashnest.com | 0.000969 | 6 | H |
| Bleutrade.com | 0.000892 | 3 | E |
| Cubits.com | 0.000489 | 1 | W+E |
| HolyTransaction.com | 0.000448 | 2 | W |
| Solo_CKPool_cb | 0.000342 | 1 | M |
| CoinSpot.com.au | 0.000208 | 1 | W+E |
| Bitcoin.de-old | 0.000197 | 2 | E |
| OKCoin.com | 0.000103 | 2 | E |
| BTCC_Pool | 0.000028 | 1 | P |
| MoonBit.co.in | 0.000018 | 1 | F |

Table 5.2. Clusters related to members of BTC.com and their corresponding percentage of BTC received and the number of addresses. Only the top 20 clusters, sorted by "% of BTC", are shown.

| Cluster | % of BTC | Tag | No. addresses |
|---|---|---|---|
| 304700830 | 13.799274 | Bixin | 1061 |
| 46553983 | 6.706924 | Huobi.com-2 | 964 |
| 327539880 | 3.395104 | None | 871 |
| 1JG9aioUDuNTcAWpSfVfoBHbJTHf5mXjsx | 2.932363 | None | 1 |
| 34HkfmwW9XdnASYBmnAEmz7XVRW37sffxK | 2.876949 | None | 1 |
| 324067473 | 2.446936 | None | 722 |
| 350822682 | 2.030124 | None | 42 |
| 350824718 | 2.029310 | None | 393 |
| 354771408 | 1.982419 | None | 2 |
| 349873556 | 1.911828 | None | 1 |
| 360509809 | 1.712476 | None | 1 |
| 372448840 | 1.502087 | None | 283 |
| 333653856 | 1.269123 | ckpool.org__cb | 435 |
| 346539747 | 1.014413 | None | 1 |
| 361611945 | 0.980146 | None | 5 |
| 331999394 | 0.780064 | None | 28 |
| 234254928 | 0.773891 | KNCMiner__ckpool | 1077 |
| 318730655 | 0.773443 | None | 1 |
| 310034330 | 0.769927 | None | 1 |

# Chapter 6

# Conclusions

With this work we provided a wide analysis of the current status of bitcoin mining. We first collected data from different sources and that we then used to extract useful information and shed some light on mining pools and pools' members. In particular, we first saw how centralized bitcoin mining is and what are the relationships between pools: For specific time periods more than 50% of the blocks have been mined by at most three mining pools, which is considered to be a threat for the protocol.

Afterwards, we analyzed how block rewards are distributed among pool members and we tried to find patterns in payout transactions. We identified some mining entities thanks to the multiple-input clustering heuristic and we found evidence of cross-pool mining from different clusters: some of them are linked to mining pools, meaning that for them it is more profitable—or at least a better return on investments—to mine within other pools, while others are linked to wallet and exchange services.

We plan to extend our pool members' analysis also to the other main pools and quantify more accurately cross-pool mining activities. Finally, in order to better analyze the mining power decentralization, it is fundamental to understand what is the percentage of mining hardware actually controlled by Bitmain and exclude the hash rate coming from other miners in their public pools. Only in this way, one can properly assess the probability of having a peer-to-peer but centralized protocol.

# Bibliography

[1] http://organofcorti.blogspot.co.at/2016/11/november-6th-2016-block-maker-statistics.html.

[2] https://bitcoin.org/en/bitcoin-core/.

[3] https://www.coindesk.com/bitmain-bitcoin-mining-launch-second-mining-pool/.

[4] https://www.crunchbase.com/organization/bitmain#/entity.

[5] https://github.com/blockchain/api-v1-client-python/blob/master/docs/blockexplorer.md.

[6] https://github.com/blockchain/Blockchain-Known-Pools/blob/master/pools.json.

[7] A. Biryukov, D. Khovratovich, and I. Pustogarov. Deanonymisation of clients in bitcoin P2P network. *CoRR*, abs/1405.7418, 2014.

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

[9] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer. Decentralization in bitcoin and ethereum networks. *CoRR*, abs/1801.03998, 2018.

[10] B. Haslhofer, R. Karl, and E. Filtz. O bitcoin where art thou? insight into large-scale transaction graphs. In *SEMANTiCS (Posters, Demos)*, 2016.

[11] 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*, IMC '13, pages 127–140, New York, NY, USA, 2013. ACM.

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

[13] K. Nayak, S. Kumar, A. Miller, and E. Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. 00:305–320, March 2016.

[14] M. Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[15] M. Rosenfeld. Summary of mining pool reward systems. https://bitcoil.co.il/pool_summary.pdf, 2011.

[16] D. Vorick. The state of cryptocurrency mining. https://blog.sia.tech/the-state-of-cryptocurrency-mining-538004a37f9b, 2018.