Analyzing the Potential Effects of Internet Censorship on the Bitcoin Network.

Jorge Coll, Andrew Fasano, Benjamin Kaiser, Lucy Qin

MIT 6.892: Shared Public Ledgers – Final Project

wehavenoprivacy@mit.edu, fasano@mit.edu, benjamin.h.kaiser@gmail.com, lucyzqin@gmail.com

Abstract—**Abstract goes here.**

I. INTRODUCTION

The Bitcoin cryptocurrency relies on a shared, global transaction ledger that each participant can agree on and independently validate. When new blocks are created that can be added to this ledger, they are broadcast over the Internet between participating nodes in a peer-to-peer protocol. If all published blocks could be guaranteed to reach every other client in the network, all clients would eventually have a consistent view of the ledger at any given point in time. However, this assumption may not always hold true. A large number of Bitcoin nodes, such as those in China, are separated from their peers by Internet censorship systems. With specific messages embedded in a block published on the blockchain, it may be possible to force those censorship systems to inadvertently prevent censored miners from viewing a block the rest of the world agrees on. We set out to investigate how Internet censorship systems could prevent Bitcoin users from reaching consensus across their borders. In particular, we analyze the feasibility and impact of an attacker embedding a malicious transaction into a block on the Bitcoin blockchain that would cause an Internet censorship system to prevent that block from being visible to Bitcoin users behind the censorship system. We examine the current implementation of the Great Firewall of China and a simulated model that drops any TCP packets that contain a censored word.

We designed a set of simulations to analyze how Bitcoin clients would behave if TCP packets containing information about a block were unable to be sent between groups of users. Our simulations also examine the long-term impact on global blockchain consensus both when a majority of miners are uncensored or when the majority is censored.

II. BITCOIN'S NETWORKING PROTOCOL

The message transfer protocol used in Bitcoin is defined abstractly in Nakamoto's original publication [1] and further fleshed out in `bitcoind`, the reference implementation he provided. In this section, we will provide a high-level overview of the relevant portion of the protocol.

The standard relay protocol used to share blocks between clients contains four types of messages:

- *inv*: When a transaction or block has been verified by a node, the node announces that the transaction or block is available by sending an *inv* message to all of their neighbors.
- *getdata*: When a node receives an *inv* message for a transaction or block that it does not already have a local copy of, it responds with a *getdata* message asking for that block.
- *tx*: When a node receives a *getdata* request for a given transaction, it responds with a *tx* message containing the transaction.
- *block* When a node receives a *getdata* request for a given block, it responds with a *block* message containing the block.

A node will only propagate a block or transaction to its neighbors if it can verify it. Unverified transactions or blocks are not propagated.

Partitions occur when two or more competing blockchain heads emerge. By itself, Bitcoin does not attempt to detect or mitigate partitions. If a partition appears due to two groups having different views of the current blockchain, both groups will continue operating independently and their states will diverge. If the divergence results in subnetworks that produce incompatible transaction histories, they will be unable to merge if the partition is later removed. Decker and Wattenhofer [2] observe that detection of network partitions may not be challenging and could be achieved through simple monitoring of the observed aggregate computational power in the network. As Bitcoin has grown in size and influence, such monitoring has begun to take place, Since at least 2013 it has been possible to view the total network hashing rate via public websites [3].

III. TRAFFIC CENSORSHIP

Censorship of Internet traffic occurs in a variety of forms all over the world. Most common is censorship of Web traffic (e.g., HTTP, DNS, and other TCP protocols related to browsing the Web). The most aggressive censorship takes place in countries where citizens' access to the Internet is very limited. Verkamp and Gupta [4] conducted experiments to infer the mechanics of various countries' approaches to Web censorship and also provided a set of independent sources ranking countries based on the severity of their censorship.

In our review of existing resesarch, four countries were repeatedly identified as the most severe censors: North Korea, Cuba, Iran, and China. Of these, China is the most interesting case for our purposes for two reasons: (1) its Internet-connected population far surpasses the others' and (2) it is a hotbed
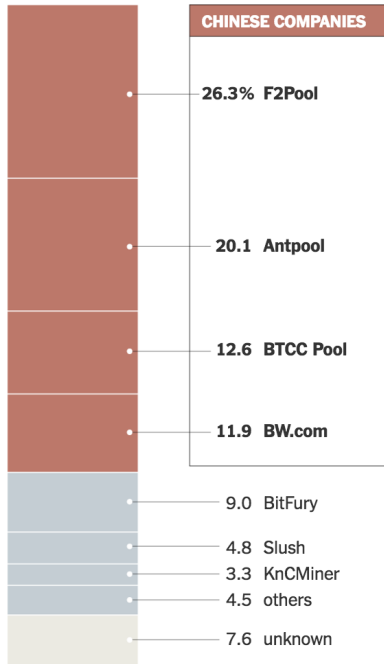
Fig. 1: From [5]: the shares of mined Bitcoin blocks from May 24 to June 24, 2016 by mining pool.

of Bitcoin mining activity, with Chinese-run mining pools accounting for an estimated 70% of all Bitcoin mining power as of June 2016 (Figure 1). As a collection, China's censorship measures are colloquially referred to as The Great Firewall of China (GFW).

IV. THE GREAT FIREWALL OF CHINA

There is no consensus about the precise technical underpinnings of the GFW, as conflicting observations have been made. In particular, there is conflict about whether or not it is stateful – i.e., whether it stores and uses information about the packets it intercepts[1]. However, according to [7] the basic mechanisms employed are known to be:

- *IP address filtering*: The GFW blocks specific IP addresses from receiving traffic by dropping all packets associated with it. This ensures that the GFW's reach extends to all content produced by a host's IP, rather than just the few specified domains.
- *DNS misdirection (hijacking)*: When requesting a blocked host name, there are cases in which the DNS servers under the GFW will return a different IP address than the one that corresponds to the domain name requested. The Chinese government can effectively replace the content on any website with material that is more favorable to their interests.
- *Keyword filtering*: If a banned keyword appears in a URL, after a completed TCP handshake, the GFW will send reset packets to both the source and destination, blocking access

[1] The most recent results, in Xu et. al[6], indicate that the GFW does record state.

to the requested content. Even if a keyword is not explicitly in the URL but appears within the HTML response, the content is also denied[2]. In this particular instance, pages often begin to display but are then truncated after the discovery of a keyword.

A. Impact on Bitcoin

Given the current implementation of the GFW, only network activity over HTTP is censored and Bitcoin activity, while slightly delayed, is not subject to filtration[3] However, this delay, combined with normal Bitcoin block propagation delays, cause Chinese miners to hear about blocks mined outside of China later than ones mined within the country, causing a communication barrier. This creates a disadvantage for miners outside of China since the majority of the hash power in Bitcoin is located inside of China. If the current proposals to increase the block size used by Bitcoin are accepted, this delay would likely increase. [8]

V. NETWORK PARTITIONING

Internet censorship techniques in use are always changing. Therefore, even though the GFW is not currently censoring Bitcoin traffic, it is conceivable that it or another censorship regime may at some point. The remainder of this paper will focus on how a network partition could be introduced into Bitcoin if a (hypothetical) censorship regime filtered valid blocks containing banned works. At a high level, the obvious result is that the valid block would be visible to the entire world except for the nodes behind the censorship system. The uncensored users would mine off the head of the blockchain while the censored users would mine off of a different block, causing a fork. Depending on the ratio of mining power between the censored and uncensored miners, the two groups may both continue mining on their forked blockchains, despite the fact that one blockchain will be shorter than the others. Until the partition is removed or circumvented, they will never be able to reach consensus.

VI. EXPERIMENTS

This section will describe the experiments we conducted. Our initial aim was to test our hypothesis by actually creating, in practice, a transaction containing a word censored by the GFW. When we determined that this was infeasible, we decided to perform a set of experiments to analyze what effects such an attack might have on Bitcoin. The first simulator we ran abstracted away too many of the important details of the Bitcoin protocol, focusing instead on collecting high-level statistics about the network. Our next step was to try to understand the expected behavior of a Bitcoin client located inside of the GFW by directly inspecting the source code of `bitcoind`. Finally, we worked to deploy and run a more detailed simulator. Unfortunately, configuration issues prevented us from running our intended simulations, so at the moment they are left as

[2] We were unable to reproduce this behavior at the time of writing.

[3] This means that Bitcoin blocks could be used as a covert channel for smuggling censored information in and out of China.

future work. However, we do feel that through this process, we developed strong hypotheses about how the network would behave in the presence of a censorship-based partition, and as future work we intend to resolve our configuration issues with the simulator and test those hypotheses.

A. Experiment Design

To test our hypothesis, we needed to conduct two experiments to understand how the the GFW blocks traffic and how the Bitcoin client behaves when certain blocks cannot be propagated between two groups of clients.

1) Firewall Testing: To check if the GFW would prevent the propagation of a Bitcoin block from crossing through its censorship systems, we obtained two machines: one in China (CN) and one in the United States (US). To confirm that our CN machine was affected by the GFW, we attempted to access several blocked websites and confirmed that they were inaccessible.

We attempted to send a variety of messages from our US machine to our CN machine and vice versa. We used the netcat UNIX utility to send ASCII messages. When we sent a message causing the GFW to block the connection, we expected the CN machine to be unable to reach the US machine on the same port for a few minutes, as described in [9].

After validating that the GFW was affecting traffic between our two machines, we sent test messages of various formats containing known, blocked words. For each type of message, we recorded if the connection was blocked.

2) Simulation: By simulating a small-scale version of the Bitcoin network with network censorship in place, we aimed to examine how a censorship-induced fork would affect the network over time. The majority of Bitcoin nodes run `bitcoind`, the reference client implementation of the Bitcoin protocol [10].

Data can be collected simply by simulating the network traffic between nodes, but a thorough study should include a simulation of an actual Bitcoin network with nodes running `bitcoind`.

Bitcoin Simulator [11] built on top of NS3 [12] simulates network traffic between bitcoin clients.

Shadow [13] running with its Bitcoin plugin [10] emulates real `bitcoind` clients in a simulated network that can be used to conduct experiments.

The possible scenarios we wanted to examine fall into four categories:

1) 51% of mining power is unaffected by censorship, meaning that 49% of mining power behind the censor cannot send/receive messages containing a blacklisted word.
2) 49% of mining power is unaffected by censorship, meaning that 51% of mining power behind the censor cannot send/receive messages containing a blacklisted word.
3) 30% of mining power unaffected by censorship with 70% unable to send/receive messages, approximately reflecting the real-world distribution of Bitcoin hash power outside and inside of China.
4) A majority of mining power is unaffected by censorship (partition $P1$), a minority of mining power cannot send/receive messages to/from $P1$ containing a blacklisted word ($P2$), and a small number of miners is unaffected by censorship and can communicate to both Partitions 1 and 2 without any censorship ($P3$).

We aimed to embed a transaction with well-documented blacklisted words through using the op code OP_RETURN, which marks the transaction output as invalid but still allows for the transaction to be propagated. There are a variety of other methods including sending a transaction to a fake address name containing a blacklisted keyword. Through the use of the coinbase field, a miner could also launch this attack by appending blacklisted keywords onto a block

B. Expected Results

Suppose we have client C, a censored miner, sitting behind a censorship firewall, a client U who is uncensored, and a block B_0 which is the last valid block known by all parties before the partition emerged. Now suppose U mines block B_1 containing a transaction with a blacklisted keyword. U transmits an *inv* message about B_1. Since C has never seen B_1, it requests the full block from U by transmitting a *getdata* message for B_1. U begins transmitting C a *block* message for B_1, but the censorship firewall notices a blacklisted keyword and intervenes. This can be seen in Figure 2. Regardless as to whether the censorship firewall truncates the transmission of B_1 from U to C or simply resets the connection, the end result is that C will never receive B_1 but will know of its existence (since it received the *inv* message about B_1).

To better understand what would happen if a node knew about the existence of a block which it could never retrieve, we examined the source code to `bitcoind`. In Bitcoin, clients learn of new blocks by listening for block header broadcasts. Once it hears of a new block header, the client attempts to download the block. The reference client keeps track of the active chain; however, it will only ever update the active chain once the block header and the full block are both received and validated. According to a comment in the source[4], this logic was introduced to defend against malicious miners trying to obtain a competitive advantage by broadcasting headers but withholding blocks. Therefore, we expect C to continue mining off B_0, the previously known block, thereby causing a fork.

1) Censored minority: Figure 3 shows the scenario in which the minority of hash power is behind the censor. Censored miners will see future blocks added by uncensored miners, but will not be able to validate their chain, so they will be ignored.

2) Censored majority: If censored miners mine faster than uncensored miners, the two forks will eventually grow to be of equal length. Censored miners will only be able to see one of the forked paths while uncensored miners will be able to see (and mine off of) both. This can be seen in Figures 4 and 5.

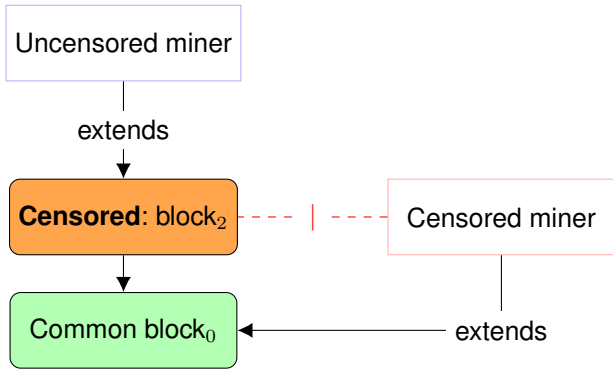[4]Specifically, line 2668 of `validation.cpp` in `bitcoind` version 0.14 [14]

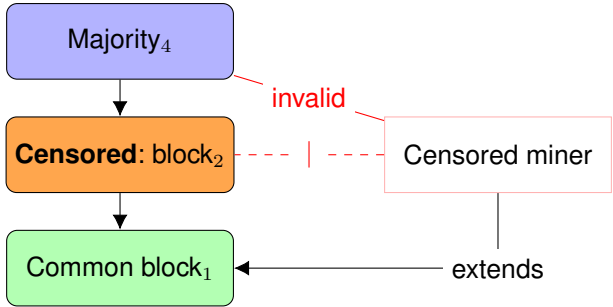Fig. 2: After a censored block is mined, miners try extending different blocks

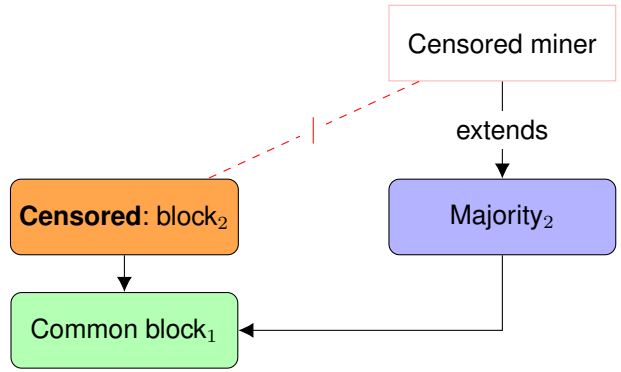Fig. 3: A model in which the minority of hash power is located behind the censor

Fig. 4: A model in which the majority of hash power is located behind the censor
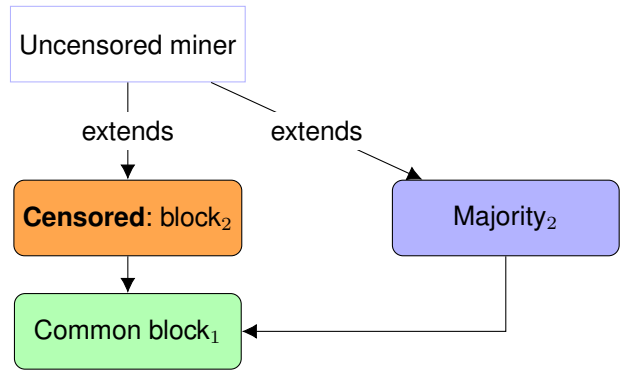
Fig. 5: Uncensored miners can mine off of either block when the forked chains are of equal length

In this scenario, one fork can be mined off of by all nodes while the other can only be mined off of by uncensored nodes. Therefore, the first fork is likely to grow more quickly than the second, meaning that it will become accepted as the main chain. This will prevent transactions containing censored keywords from ever being accepted into the main chain (Figure 6).

C. Results

1) Firewall Testing: To test the GFW we first had to obtain access to a system with a censored Internet connection. When a message caused the GFW to block a connection, the connection was reset and the CN machine was unable to connect to the US machine on the same port for a few minutes.

We attempted to construct a Bitcoin block that would be filtered by the GFW. Arbitrary data has been embedded into blocks before; the most common places that such data is stored in a block are the address fields (by encoding the data in hex), the coinbase transaction, or the scripts[5].

In the end we determined that we would need to trick the GFW into interpreting a block as Web traffic of some sort (e.g., DNS or HTTP), which was not feasible.

2) Simulation: Initial analysis of the Bitcoin Simulator built on NS3 showed that the simulator didn't simulate at

a high enough level of fidelity for us to conduct our desired experiments. Instead, we focused on using Shadow.

We compiled version 0.9.2 of the `bitcoind` client[6] for use in Shadow. The published instructions for running Shadow's Bitcoin Plugin were significantly outdated, so we created a Docker container that builds their simulator and applies patches so that their original experiment can be recreated[7]. In our Docker container, we ran a network of 4 bitcoind clients communicating as we broadcast transactions and blocks to one of the clients.

We patched the `bitcoind` client to drop messages it receives containing a "blocked" keyword.

We reconfigured Shadow to run a network with two groups of two nodes that were unable to communicate with each other. We then sent blocks to only one of these groups, expecting the clients in that group to have a different blockchain than the clients in the other group.We were unable to configure Shadow to run in this fashion due to time limitations.

If we were to get Shadow to separately partition two network segments, we would have then modified how Shadow sends

---

[5]In one notable case, a simple cross-site scripting (XSS) attack was embedded in a block and was apparently demonstrated to work on blockchain.info[15], although the site has since been patched to properly escape HTML.

[6]Shadow required us to use a slightly modified version of 0.9.2 available at https://github.com/amiller/bitcoin/tree/0.9.2-netmine

[7]The Dockerfile that builds this container is available at https://github.com/AndrewFasano/shadow-bitcoin-docker
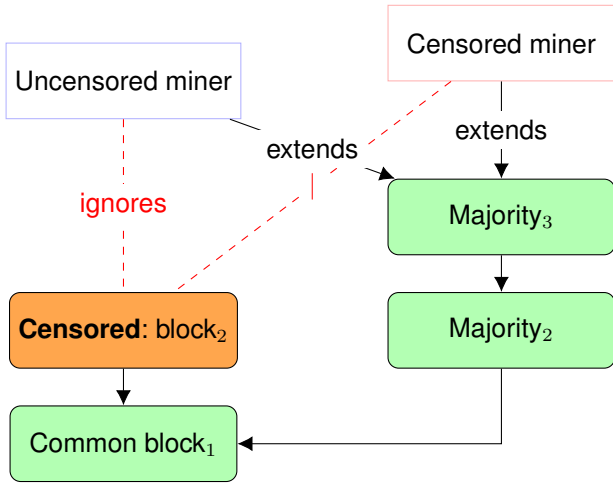
Fig. 6: Since the majority of mining power is mining off one fork, that fork will eventually grow to be longer than the other and all miners will agree on it

messages between clients to drop any messages between the two groups where the message contains our blocked keyword.

VII. LIMITATIONS

Speaking broadly, experiments performed on a model are only useful to the degree to which the model accurately represents reality. Therefore, we want to make explicit the limitations of our model and the simplifying assumptions that we made when designing our simulations. These limitations are as follows:

- *Mining pools*: In practice, mining pools in which miners cooperate to find new blocks and share the rewards are common [16]. We model mining pools as a single miner who has the cumulative hash power of the pool. This fails to account for scenarios in which miners outside of the GFW contribute to a pool that is managed by a node inside of the GFW.
- *Miner intervention*: If the blockchain were to fork due to a mined block that exploits a censorship system, Bitcoin users would certainly notice and possibly attempt to rectify the situation. In attempt to stabilize the currency, it is possible that a large group of miners could quickly react to a fork and behave in an unexpected way. For instance, in a scenario where a majority of mining power is uncensored, an uncensored group of miners pool could modify their clients to ignore blocks that would be censored. If by doing this, they shift a majority of mining power to behave as if it were censored, the entire Bitcoin network will eventually reach agreement on a blockchain that does not include the malicious block.
- *Non-Standard Clients*: While [10] shows that the Bitcoin reference client is used by a majority of Bitcoin users, it does not specify the percentage of mining power that is using the client. This attack would not have the same affect on miners using non-standard clients that do not

require validation of blocks or use a different protocol or encoding to share blocks.

VIII. DISCUSSION

Although we do not believe Bitcoin is presently censored by nation states, we believe it could be a subject of scrutiny in the future. Given that the majority of hash power resides in one country, the conscious decision to censor Bitcoin activity could have detrimental effects to the network as a whole. What originated as a decentralized currency has manifested in a concentration of power in one geographic region. This leaves the network vulnerable to decisions made by regimes that can choose to block not just blacklisted keywords but also legitimate payments in a variety of methods based on the attack we described.

Thankfully, censoring every device within a geographic region is a big challenge. Multiple ways around censorship firewalls have been proven to work in the past; e.g., web proxies, VPNs, Tor, etc. [citation needed]. Bitcoin nodes nowadays even resort to using a variety of protocols to communicate block information, such as Bitcoin FIBRE [17] and Compact Blocks [18]. Fully blocking communication between miners on two sides of a censorship wall is still a difficult feat as censored nodes would only need one point of communication that can bridge the two sides. Of course, this would be at the cost of added network latency, which has economic and network implications on its own. If this attack were to come to fruition, the organization of mining pools could help mitigate against its effects. If the pool manager were to relocate to another location that is not subject to censorship, the manager could still select blocks and relay them back to miners located within the firewall.

Given that Bitcoin is not currently subject to censorship, this creates an opportunity for individuals to add blacklisted messages onto the blockchain. Though there are already many tactics to circumvent the censorship of information, the blockchain could provide an additional outlet.

IX. FUTURE WORK

Now that we have created a reproducible system to run Bitcoin under Shadow, we plan to use this system to conduct the experiments we have described. Our source code analysis used to form our expected results leaves us confident that the simulation will behave similarly to what we anticipate. However, by running the real Bitcoin client in this simulation, we will be able to confirm this hypothesis and also examine any potential side effects caused by clients repeatedly requesting blocks from other miners but never receiving the blocks.

In this research, we focused exclusively on Bitcoin and the GFW and did not examine any other real-world cryptocurrencies or censorship regimes. Some permutation of these may be currently vulnerable to the attack we described.

THANKS

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system."

[2] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing*, 2013.

[3] [Online]. Available: http://bitcoin.sipa.be/

[4] J.-P. Verkamp and M. Gupta, "Inferring mechanics of web censorship around the world," in *Presented as part of the 2nd USENIX Workshop on Free and Open Communications on the Internet*, 2012.

[5] N. Popper, "How china took center stage in bitcoin's civil war," The New York Times, June 2016.

[6] X. Xu, Z. M. Mao, and J. A. Halderman, "Internet censorship in china: Where does the filtering occur?" in *Proceedings of Passive and Active Measurement: 12th International Conference*, 2011.

[7] M. Hu. (2011, May) The great firewall: a technical perspective. Torfox: A Stanford Project. [Online]. Available: https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/FreedomOfInformationChina/great-firewall-technical-perspective/index.html

[8] [Online]. Available: http://www.nasdaq.com/article/why-the-great-firewall-of-china-is-causing-serious-issues-for-bitcoin-miners-cm585604

[9] R. Clayton, S. J. Murdoch, and R. N. Watson, "Ignoring the great firewall of china," in *International Workshop on Privacy Enhancing Technologies*. Springer, 2006, pp. 20–35.

[10] A. Miller and R. Jansen, "Shadow-bitcoin: Scalable simulation via direct execution of multi-threaded applications," in *8th Workshop on Cyber Security Experimentation and Test (CSET)*, 2015.

[11] [Online]. Available: https://arthurgervais.github.io/Bitcoin-Simulator

[12] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *Modeling and tools for network simulation*, pp. 15–34, 2010.

[13] R. Jansen and N. Hopper, "Shadow: Running tor in a box for accurate and efficient experimentation," in *Proceedings of the 19th Symposium on Network and Distributed System Security (NDSS)*. Internet Society, February 2012.

[14] [Online]. Available: https://github.com/bitcoin/bitcoin/blob/0.14/src/validation.cpp#L2668

[15] [Online]. Available: https://redd.it/1n57uj

[16] M. Rosenfeld, "Analysis of bitcoin pooled mining reward system," 2011.

[17] [Online]. Available: http://bitcoinfibre.org

[18] [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0152/protocol-flow.png