IB Extended Essay

Computer Science: How vulnerable is the Bitcoin network to Subversive Mining attacks?

Benjamin Congdon

Candidate Number: 000944-0023

Computer Science Extended Essay

Word Count: 3998 words

Advisor: Tom Donnelley

Session: May 2015

Abstract

This essay seeks to answer the research question: *How vulnerable is the Bitcoin network to Subversive Mining attacks?* An investigation of the current protocols of the Bitcoin network was conducted with specific focus on the topics of fork resolution and node-to-node interactions, as they were particularly pertinent to considering vulnerability of the network. Following this, a variety of sources were consulted to determine the 'procedure' and feasibility of such attacks - with specific mention to Block Discarding attacks and Selfish Mining attacks because of the prevalence of relevant literature. Additionally, several methods of altering the current Bitcoin protocols were evaluated for their merits in closing vulnerabilities in the behavior of individual nodes. This was seen as relevant to the research question because Bitcoin is in continuous development, so the simplicity of implement countermeasures against subversive mining directly relates to the long-term vulnerability (or lack thereof) of the network.

The conclusions of this essay were that under current conditions, the Bitcoin network is at substantial risk for a subversive mining attack. Node behaviors as dictated by network protocols do not incentivize 'honest' practices; rather, there is an incentive to withhold information, putting the attacker in an advantageous position to receive mining rewards - disproportionate to the amount of 'work' they contribute. The lower threshold for the amount of computational power is nearly zero, as contemporary methods of fork resolution favor the party with fastest block propagation. Despite these troubling findings, methods to raise the minimum threshold of profitability for selfish mining were found to be simplistic, generally effective, and able to be retroactively implemented. Thus, while the network is currently at substantial risk, given responsible maintenance of the network, it is likely that in the long term there is little risk of a destabilizing Subversive Mining attack.

Word Count: 294

Table of Contents

List of Figures

1 Introduction

1.1 Research Question

How vulnerable is the Bitcoin network to Subversive Mining attacks?

1.2 Bitcoin Cryptocurrency

The Bitcoin cryptocurrency was first envisioned by Satoshi Nakamoto in 2008, revolutionary for its use of a proof-of-work chain to verify transactions in a decentralized way (Nakamoto, 1). In this fashion, Bitcoin attempts to offer an alternative to traditional government-backed currencies using cryptographic proof and self-regulating protocols. The currency has seen an explosion of growth; as of October, 2013 the total market capitalization of Bitcoin was estimated at $1.5 billion USD (Eyal, 1). As there is no intrinsic value of Bitcoins, much research has gone into the security of the Bitcoin protocol; traditionally this has been focused towards the feasibility of History-Revision style attacks. This type of attack was theorized as early as the creation of Bitcoin itself, but is infeasible to the point of dismissal because the attacker needs to control more of the network than the 'honest' sect of the network (Nakamoto, 4). Bitcoin shows promise to become a legitimate alternative to traditional currencies, so analysis of its potential vulnerabilities is an imperative of ensuring its future growth and acceptance.

1.3 Methodology

This essay will explore the feasibility and threat of a 'subversive mining' attack, which takes advantage of the incentive-based mining system to allocate a disproportionately high amount of rewards to the attacker. This group of attacks, comparatively newer than History-Revision attacks, theoretically necessitates much less network superiority to be performed (Bahack, 1). Subversive mining operates on the principles of node collusion and network penetration. In short, Bitcoin relies on an honest network of nodes that independently follow the behaviors prescribed by the Bitcoin protocol, often without 'proof' that these protocols are being followed. As such, vulnerability arises out of emergent behavior that allows attackers to have a statistical advantage, not out of clearly defined exploits in network protocols. Therefore, this type attack is currently theoretical, and as such must be evaluated by consulting and applying current research into the feasibility of such attacks. An analysis of current protocols and structures within the network will be used to determine the extent of Bitcoin's immediate and long-term vulnerability to subversive mining.

1.4 Bitcoin Network Framework

To understand the potential risk of subversive mining, one must first have a knowledge of the 'honest' protocols prescribed by the Bitcoin network. Bitcoin uses a combination of public-key cryptography and hashing algorithms to create a system by which anyone can verify the ownership of currency and the validity of transactions (Nakomoto, 2). This creates a decentralized currency backed by cryptographic proof rather than trust in an authoritative entity.

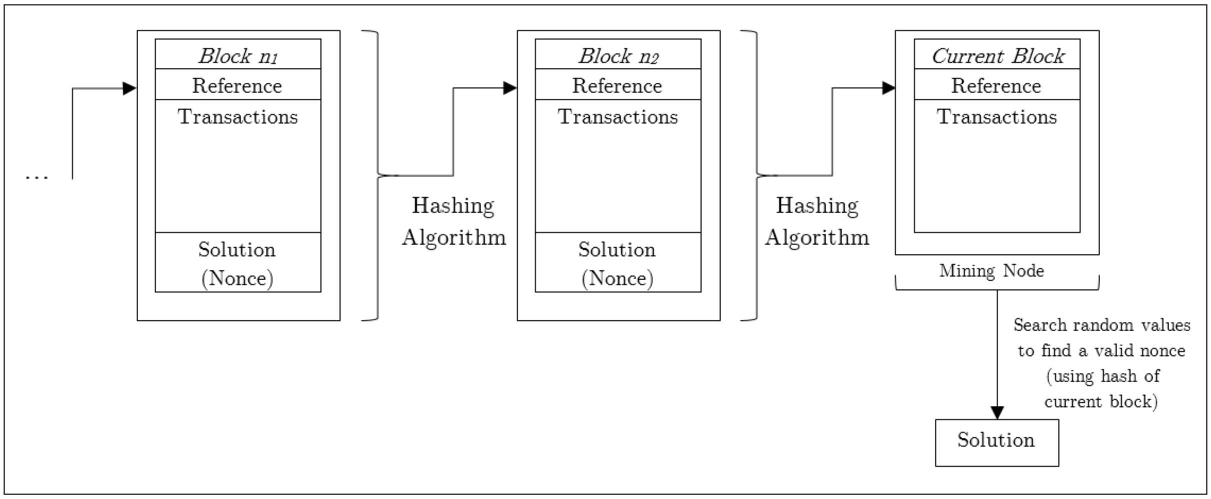
1.4.1 Transaction Structure

Bitcoins are traded in transactions wherein the sender 'signs' away an amount of Bitcoins with his/her private key and the public key of the receiver, referencing previous transaction(s) to prove they own the currency (Courtois, 3). This method of trading currency verifies ownership of currency (in that anyone with the knowledge of a payer's public key can verify the cryptographic validity of their transaction), and links transactions together so that each coin can be traced back to the genesis of that coin, verifying it as valid currency. However, this system does not verify that the payer has not already spent that currency, just that it was controlled by them at some point. This problem is known as the double-spending problem, in that a system without authoritative timestamps cannot guarantee that one sender has not sent the same Bitcoins to two recipients, resulting in ambiguity as to who actually 'owns' the currency (Bonadonna).

1.4.2 Blockchain Structure

To combat double-spending, the Bitcoin network arranges transactions into time-stamped blocks which, like transactions, are linked together in a chain, each containing a reference to the previous block (Ober, 238). As each block is linked to the previous block, this chain allows the network to agree upon the order of transactions, ensuring that no double-spending occurs. Blocks are created by members (nodes) of the Bitcoin network searching for answers to a cryptographic puzzle which, when found, serve as proof-of-work, allowing for the creation of a new 'mined' block (Nakomoto, 3). The proof-of-work secures the blocks into the chain, as recreation of a block would require the same amount of work to be done. This becomes increasingly difficult as the chain grows in length away from the block one wishes to reproduce. In another sense, this proof-of-work ties real time to computational time, allowing for a distributed method of time stamping transactions.

Figure 1: Structure of the Bitcoin Blockchain

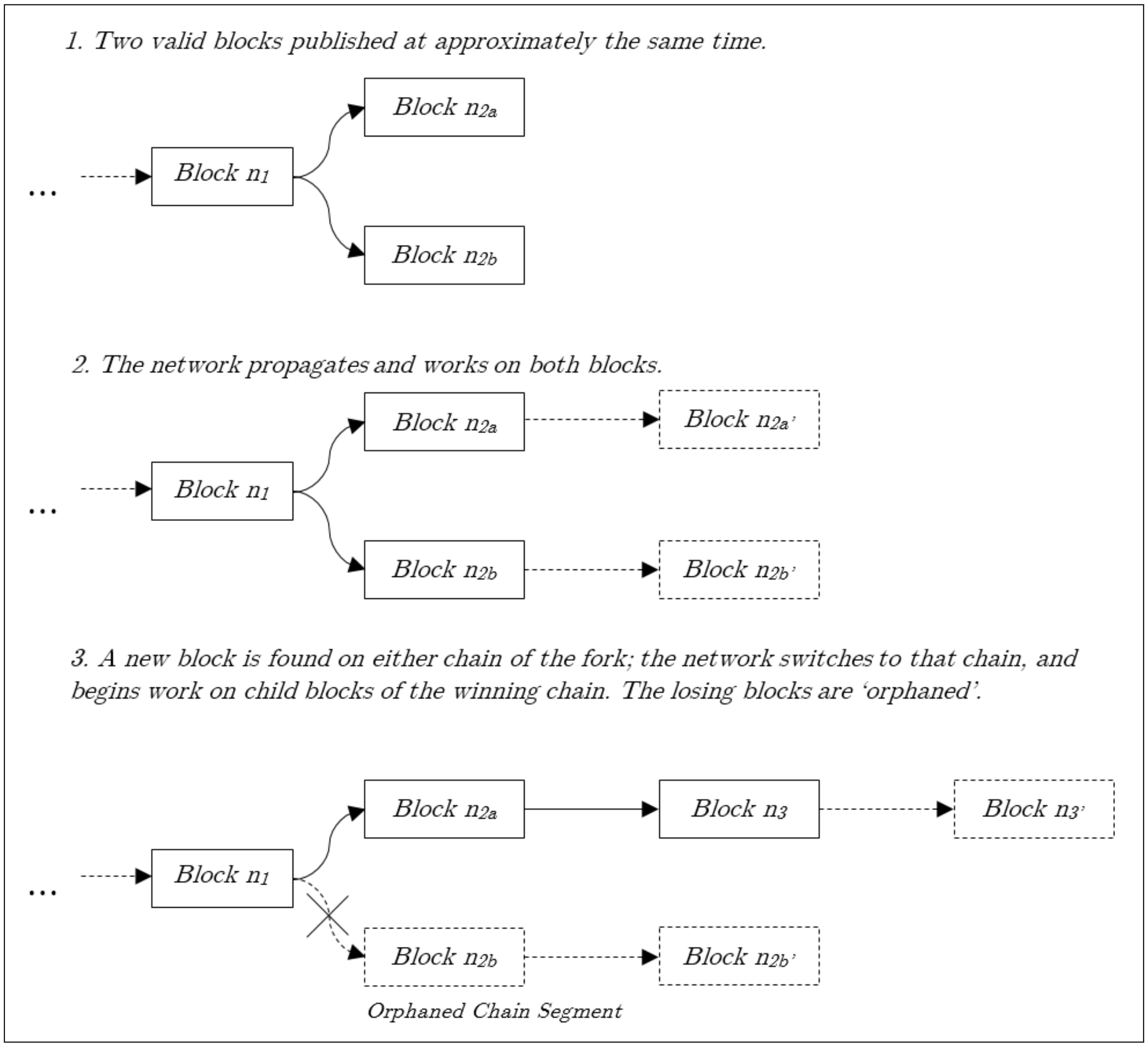


The proof-of-work puzzle used to create and link blocks utilizes hashing algorithms because of their random characteristics and necessity of computational time. Hashing algorithms are essentially a form of one-way encryption in that for a given input, they will always produce the same output, but given an output (the 'hash') there is no way to discern what the input was; by this same merit, the output of a hash is difficult to predict as slight changes in the input create random variation in the output (Savage, 5). The 'cryptopuzzle' that nodes work to solve involves putting the hash of the previous block, the transactions within the current block, and a random value through a hash function to try to find an output value with a desired number of leading zeros (Courtois, 3). The overwhelming majority of random values, of course, do not produce valid solutions; when a solution is found, a new block is created. As all this information, including the random value used to solve the proof-of-work, is public, it is possible for any node in the network to independently verify any block, including ones they themselves did not create. It is also worth noting that the verbiage used to describe this process, that nodes 'work' to mine towards blocks, is a slight misnomer; nodes do not make iterative or procedural calculations towards the finding of new blocks, rather they 'roll' random numbers until a block solution is found.

1.4.3 Network Structure

The final elements of the Bitcoin network are the interaction between nodes and the incentive to find proof-of-work solutions. When a node finds the solution to a block, it broadcasts this solution and begins to work on the next block; the node's neighbors verify the solution as correct, then broadcast and begin to work on the next block, and so on until all nodes on the network agree of the blocks creation (Eyal, 6). Utilizing a peer-to-peer broadcasting structure allows the network to base agreement on a decentralized system of implicit majority agreement, rather than the decision of a single authoritative entity. Should two valid blocks be discovered at roughly the same time – within the window of block dissemination – the current Bitcoin protocol instructs nodes to work on the first block they receive, and will only switch to the other 'fork' in the chain if it becomes longer than the one they are working on (Barber, 5). This is a contentious portion of the Bitcoin that will be expanded upon later. As implemented, the block that is 'heard-of' faster logically has a higher chance of being accepted in the chain because more miners will build on top of it until it becomes the longer chain, resulting in all miners switching to that chain, implicitly resolving the fork. Blocks which are not built-upon are discarded, meaning that the computational power required to mine them was wasted (Barber, 5). Thus, the P2P network provides fluid means of ensuring agreement on transaction order using a proof-of-work chain and decentralized fork resolution techniques.

Figure 2: Illustration of a Fork


1. Two valid blocks published at approximately the same time.
...
Block n_1
Block n_{2a}
Block n_{2b}
2. The network propagates and works on both blocks.
...
Block n_1
Block n_{2a}
Block $n_{2a'}$
Block n_{2b}
Block $n_{2b'}$
3. A new block is found on either chain of the fork; the network switches to that chain, and begins work on child blocks of the winning chain. The losing blocks are 'orphaned'.
...
Block n_1
Block n_{2a}
Block n_3
Block $n_{3'}$
Block n_{2b}
Block $n_{2b'}$
Orphaned Chain Segment

Finally, to incentivize miners for their efforts in producing proof-of-work blocks, every found block creates a designated amount of Bitcoins in the ownership of the miner who created the block (Nakamoto, 4). This action can be seen to serve two purposes: to reimburse miners for the electricity and hardware costs of supporting the network, and to introduce new currency into the system. Thus, the incentive is to contribute blocks to the block chain as fast as possible, as only when a block is accepted into the chain are miners actually 'paid' for their work. Under honest protocols, there is healthy competition to create these blocks, but this improves the health of the network as it is maintained by an increasingly diverse set of miners.

To summarize, the structure of Bitcoin can be illustrated as a three-tiered system comprised of the following elements:

- *Transaction Chain*: Each Bitcoin can be traced back to its genesis as a reward for a mined block by way of the cryptographic signing method used to verify transactions.
- *Block Chain:* Transactions are grouped into blocks, giving a widely-accepted ordering transactions, making double-spending impossible.
- *Peer-To-Peer Node Network*: Nodes individually work towards solving new blocks, verify the validity of transactions, and disseminate found blocks to other nodes in the network. Miners are compensated for their work by receiving payment for created blocks.

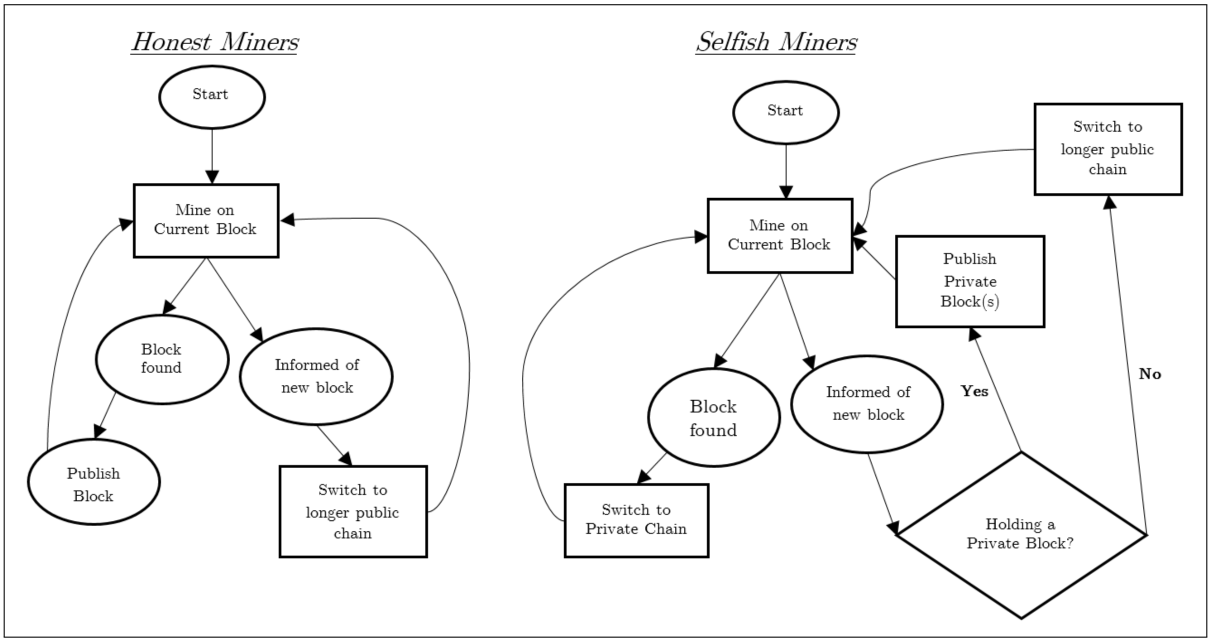
2 Principles of a Subversive Mining Attack

For the purposes of this essay, 'subversive mining' will be defined as any action taken by a Bitcoin node or group of nodes that places them at an unfair advantage to find a find a new block, or be attributed to the finding of a block. The core principle of the subversive mining attack is that, counterintuitively, there is a quantifiable advantage for the finder of a block to not publish their block immediately after finding it (Courtois, 6). Recall that for a miner to receive their payout, the network must collectively decide that the miner's block is correct and build upon that block for it to become secured in the block chain. Conventional logic would dictate that it is in a miner's best interest to publish their block instantly, to minimize the chance of losing revenue to a competitor. However, given that a miner who has found a block can instantly work on the next block in their chain, withholding the mined block allows them to get an early start on the next block. Assuming a competing block is not found, this behavior effectively wastes the hash power of the rest of the network, and gives the attacker a competitive advantage.

2.1 Block Discarding Attack

The most basic type of subversive mining is a Block Discarding attack. In this attack, subversive nodes hold onto found blocks until the instant they hear of a competing block, at which point they publish their block to the network (Bahack, 2). This action creates an intentional fork in the block chain, which as alluded to early, is solved by nodes mining on the first block of which they become aware. So, for the next iteration in the block chain, some portion of the network will work on the 'honest' block and some portion will work on the attacker's block. As the attacker had an early start, they have the advantage to find the next block on their fork, thereby discarding the 'honest' block as the network moves to consensus on the longer fork. Of course there is a distinct chance that the honest block will be favored over the attacker's block, but regardless of the result the outcome is a discarded block. This stands to waste a significant amount of the network's computational power because of the consistent lack of consensus due to perpetual forking.

Figure 3: Comparison of Honest and Subversive Mining Practices



Discarded blocks have various other adverse effects on the stability of the network. The network auto-adjusts the difficulty of finding blocks to the block-finding rate of the network so that, by convention, blocks are found at an average rate of one every ten minutes (Brezo, 20). Additionally, as blocks serve to validate transactions, a slower, more indecisive network would result in higher processing times from when a transaction is announced to when it is verified in the block chain.

2.2 Selfish Mining Attack

The practical difference between a Block Discarding Attack and a Selfish Mining Attack is somewhat pedantic, but the distinction marks a clear contrast in methodology for initiating the attack. Whereas in Block Discarding, the attacker's goal is to make as many forks as possible (slowing down the network and wasting resources), a Selfish Mining Attack uses a similar technique in an attempt to maximize mining revenue. Using the same technique of block withholding, the attacker to win an advantageously high number of blocks in proportion to their percentage of hashing power of the network as a whole, resulting in a disproportionate amount of the mining rewards allocated to the attacking group. Thus, the strategic advantage of holding the private chain is getting a 'head start' in mining the next block, not causing the rest of the network to waste computational time. The only protocol difference is that Selfish Miners continue mining on their privately held block (before publication), creating a private chain. Infrequently, this means that several blocks will be found on this chain before the honest network catches up and these blocks are published. Thus, the goal is to produce the largest amount of legitimate blocks – increasing revenue – not to create as many forks as possible.

3 Feasibility

As the Bitcoin network currently operates, the threshold of network control necessary to carry-out a subversive mining attack is essentially zero, suggesting that any single selfish miner would have a theoretical advantage (Eyal, 2-3). Practically however, a solo miner would not expect to make substantive gains from hiding their found blocks because the attack 'bets' on the attacker being able to make substantive gains towards finding another block. As there is a risk in losing blocks due to competition, the attacker must ensure that the advantage of an early start outweighs losses due to competing blocks. In essence, any miner can mine selfishly, but those with larger control of the network will benefit more from withholding blocks. Therefore a critical component of getting returns from subversive mining is collusion between miners, both for the purpose of additive hashing power and faster block dissemination of the attackers' blocks.

3.1 Pool Behavior

Collusion in the bitcoin network is not inherently a threat to the safety of the network, as mining pools are an accepted structure within the network. A 'pool' is a collection of miners that search for blocks as a group and distribute the revenue from finding blocks amongst its members, usually in proportion to the hashing power contributed by each member (Brezo, 2). Logically, these groups of miners find blocks more often than lone miners because of their larger combined hashing power, resulting in more often, but smaller payouts.

Within the context of subversive mining, colluding pools become a powerful candidate for performing an attack. Recall that as a block is broadcast along the P2P network, each node independently checks the block for validity, before beginning work and sending the block to its neighbors. Assuming this verification is not provable, it is possible for a node to skip this verification process for certain blocks, causing these chosen blocks to be disseminated in the network faster and thus have an advantage (Bahack, 2). A colluding pool could in theory choose to rapidly spread its own blocks while maintaining fair behavior on blocks it did not find, creating an advantage to its found blocks in case of a fork in the block chain. Therefore, there is an incentive for pool collusion as higher pool payouts merit higher payouts for members of that pool.

3.2 Profitability

A critical component of analyzing the practical feasibility of a subversive mining attack is its real-world profitability. While the threshold for a theoretical attack is near zero in the current protocol, an attack is only really successful it is profitable to the attackers, meaning they win more mining rewards than their hashing power would 'fairly' get them. Given α is the proportion of entire mining power controlled by the attacker, and γ is the proportion of the mining population that chooses to work on the attacker's block, the following inequality represents the range of profitability:

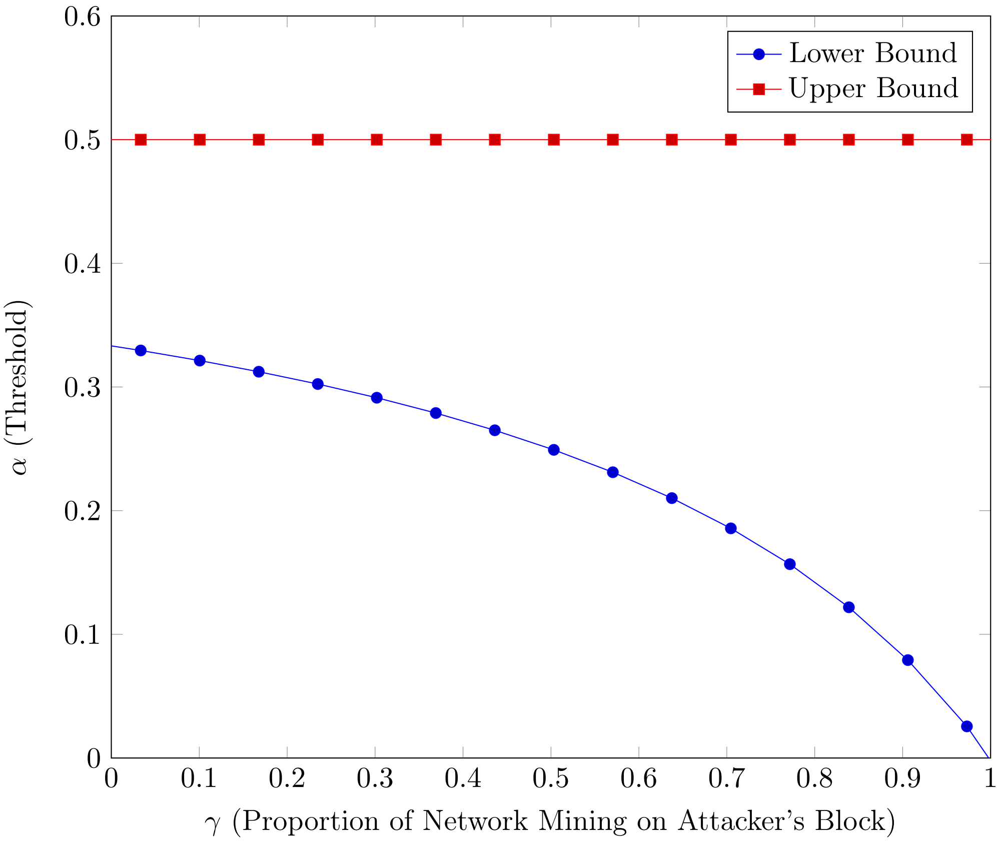
$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2} \qquad \text{(Eyal, 11)}$$

Within the current protocol, γ fluctuates between 0 and 1 due to the peer-to-peer structure of the Bitcoin network. Assuming the attacker is a malicious pool and can therefore garner higher γ

values with the influence of its members, a γ of close to 1 can be plausibly achieved. In this case, the lower boundary for α is nearly zero. Of course, values of γ are often significantly less than 1, resulting in a higher threshold, because with less of the network mining on its block the attacker needs to have greater proportion of the total network hashing rate to stay profitable. It is of note that the cap on the size of the subversive mining pool size is $\alpha = \frac{1}{2}$ as any pool that controls more than half of the hashing power of the entire network essentially decentralizes it, putting it at risk of a history-revision attack.

Figure 4: Mining Proportion Threshold of Current Fork Resolution Method

Modeled by Equation from (Eyal, 11)



4 Countermeasures

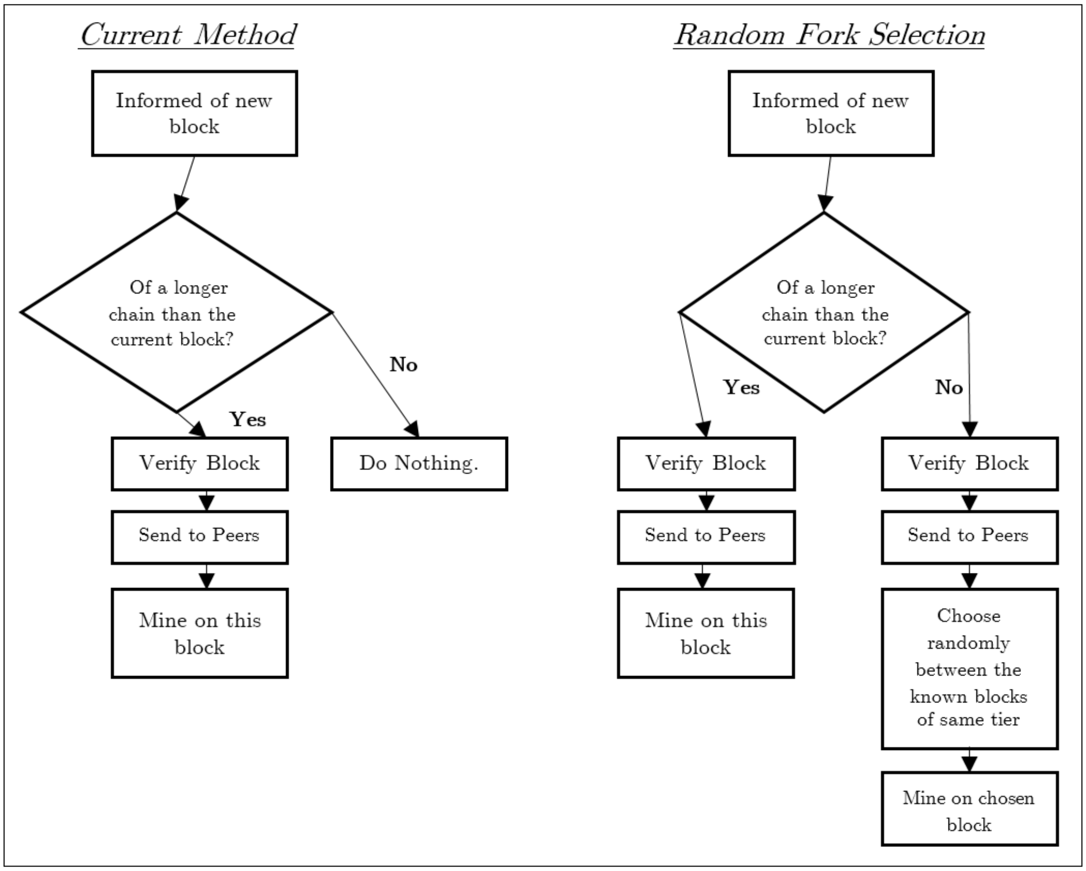
With this information in mind, the troubling implications of subversive mining attacks becomes clear; currently, at essentially any hashing speed, it can be more profitable to mine selfishly than it would be to abide by the 'fair' protocols (depending on the attacker's ability to quickly spread their block). Assuming it is impossible to prevent nodes from keeping information private, thus creating the conditions for a selfish miner, efforts to raise the difficulty of such an attack must focus on lowering the average value of γ, so as to increase the threshold of mining power needed to carry out an attack. In order to fix the current network vulnerabilities, a solution must either create a disincentive for selfish mining or make hiding information from the network more difficult.

4.1 Random Fork Selection

The concept behind Random Fork selection is that upon the discovery of a fork in the blockchain – which is to say, a node receives two valid blocks in the same tier of the chain – the node relays all competing blocks to its neighbors, and chooses randomly amongst the competing blocks which fork to work on. By randomly choosing between the honest miner's and the attacker's block, the average γ is 0.5, making the lower α threshold 0.25 (Eyal, 13). As this change in node behavior only changes the logic within a node, and not the structure or methods of communication between nodes, this measure could be introduced retroactively, without needing the entire network to switch to a new protocol. This solution diminishes the threat of a colluding pool 'flooding' the network by instructing its nodes to skip block validation on their block, as randomly selecting a fork does not give an definite advantage to the block that propagates faster.

Assuming network wide acceptance of this policy, there would still be an advantage to the block that wins the dissemination race; however, unlike with the current system, each node that is informed of a block – whether it be the attacker's or an honest block – does not immediately become a 'vote' for that party. Under this system, even if an attacker can instantly transmit their block to all nodes when they hear of an honest block, half the network would eventually settle on mining each fork, setting the chance of each party winning at 50%. (Within this discussion, it is worth mentioning that the 50% of nodes working on a given block is technically not same as 50% of the hashrate of the network working on each fork, as hashrates vary greatly amongst nodes. This analysis will assume the effects of hashrate distribution are negligible as each node has the same probability of choosing either fork.)

Figure 5: Comparison of Contemporary Fork Resolution and Random Fork Selection



A profitability threshold of $\alpha = 0.25$ is sufficiently high to dissuade solo miners and moderately sized pools against mining selfishly; however, the threat of an attack by a large pool would remain unchecked. As of September 28, 2014, there is still a pool – the "Discus Fish" pool – that would have sufficient hashing power to mine selfishly at a profit ("Hashrate"). Furthermore, this countermeasure would increase the indecisiveness of the network. For a fork to be created in the current framework, competing blocks have to be released at essentially the same time. However, with Random Fork Selection, at presumably any point in the mining of the 'current' block – a period of approximately 10 minutes – a competing block could be released, and half of the network would switch to it, away from the mature block. As such, this method does not maintain the incentive for nodes to always mine on the current block, as the 'point-of-no-return' for a block being accepted into the block chain is comparatively much longer.

Regardless of this shortcoming – which could be easily solved by capping the time period in which nodes would switch to a random fork – this method's simplicity and backwards compatibility lend itself to being a strong candidate for future implementation.

4.2 Freshness Preferred

The Freshness-Preferred method of solving the selfish mining model is to link blocks to real time using ‘randomness beacons’ to prove a block’s age. Blocks with newer timestamps would be given priority over older blocks. This is so that old blocks (presumably hidden by the attacking miners) will be less likely to be accepted than newly created blocks. The values from ‘randomness beacons’, which generate random strings on a consistent interval, would be hashed into the header of the block itself, so as to tie the creation of the block to a specific point in time; this process raises the threshold of α to 0.32 (Heilman, 3). It becomes more difficult for subversive miners to successfully hide their private chain as upon its introduction to the network, honest miners will shun this older chain in favor of the newer honest chain. Preferring ‘fresh’ blocks to old ones creates an incentive for miners to publish their blocks upon discovery, which addresses the root of the selfish mining problem.

It is of note that this method is fundamentally flawed in that it essentially decentralizes the network. The purpose of the proof-of-work chain is to use computational time to create a trust-free, network verifiable transaction history. While using timestamps from randomness beacons would still be verifiable, the network would be fundamentally dependent on trusting the true randomness and pre-image resistance – that is, the inability of a user to predict future values – of these beacons. It has been postulated that even with using malicious beacons – or if the timestamps attached to blocks could be faked – the threshold of a selfish mining attack would not decrease below levels of the thresholds of the current system (Heilman, 4). However, on a fundamental basis utilizing authoritative timestamps is not in parity with the rest of the Bitcoin protocol. While the technique of preferring newer blocks to older ones does address aspects of the subversive mining vulnerability, the particular implementation of Freshness-Preferred is not an appropriate solution.

5 Conclusion

In conclusion, in the present Bitcoin network protocols there is a non-negligible risk of a subversive mining attack because the threshold of feasibility is currently quite low. Moreover, the threshold for profitability in case of a selfish mining attack is low enough that moderate to large sized pools could, in theory, obtain more mining rewards than their proportion of the network hashing power would normally award them. The damage of such an attack would, in theory, slow network processing times and waste considerable mining resources on blocks that are destined to be orphaned. This vulnerability is not part of the intrinsic nature of peer-to-peer cryptocurrencies, but rather is due to block propagation and fork resolution behaviors of individual network nodes. When this behavior is compounded across the network, the possibility of a subversive mining attack arises. Thus, while the network is currently at a substantive risk of such an attack, the methods by which to close this vulnerability are relatively simple. It is unlikely that there is a way to cryptographically prove that nodes are not hiding unpublished blocks; however, by removing the monetary incentive for attackers to do so, the risk of subversive mining is mitigated. Random Fork Selection appears to be a strong candidate for a retroactive fix for flawed node behavior, with elements of Freshness-Preferred fork resolution also showing promise. Both of these methods could be implemented without considerable disruption to the protocols of the Bitcoin network, and would substantially decrease the risk of a subversive mining attack. With proactive development of the Bitcoin protocol to reform fork resolution and block propagation, the feasibility of a subversive mining attack will be significantly reduced.

Works Cited

Bahack, Lear. "Theoretical Bitcoin Attacks with Less than Half of the Computational Power (draft)." *Cornell University Library.* Cornell University, 25 Dec. 2013. Web. 26 Aug. 2014.

Barber, Simon, Xavier Boyen, Elaine Shi, and Ersin Uzun. "Bitter to Better - How to Make Bitcoin a Better Currency." *Financial Cryptography* 7397 (2012): 399-414. *Applied Crypto Group.* Stanford University, 2012. Web. 26 Aug. 2014.

Bonadonna, Erik. "Bitcoin and the Double-Spending Problem." *Course Blog for INFO 4220.* Cornell University, 29 Mar. 2013. Web. 28 Aug. 2014.

Brezo, Félix, and Pablo G. Bringas. "Issues and Risks Associated with Cryptocurrencies Such as Bitcoin." *Think Mind.* Proc. of SOTICS 2012 - The Second International Conference on Social Eco-Informatics, Venice, Italy. 21 Oct. 2012. Web. 26 Aug. 2014.

Courtois, Nicolas T., and Lear Bahack. "On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency." *Cornell University Library.* Cornell University, 6 July 2014. Web. 27 Aug. 2014.

Eyal, Ittay, and Emin G. Sirer. "Majority Is Not Enough: Bitcoin Mining Is Vulnerable." *Cornell University Library.* Cornell University, 15 Nov. 2013. Web. 27 Aug. 2014.

"Hashrate Distribution." *Blockchain.* Blockchain, 28 Sept. 2014. Web. 28 Sept. 2014. <https://blockchain.info/pools>.

Heilman, Ethan. "One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner." *Cryptology EPrint Archive.* International Association for Cryptologic Research, 4 Apr. 2014. Web. 2 Sept. 2014.

Nakamoto, Satoshi. "Bitcoin: A Peer-to-Peer Electronic Cash System." *Bitcoin.* Bitcoin Foundation, 31 Oct. 2008. Web. 25 Aug. 2014.

Ober, Micha, Stefan Katzenbeisser, and Kay Hamacher. "Structure and Anonymity of the Bitcoin Transaction Graph." *Future Internet* 5.2 (2013): 237-50. *Future Internet.* MDPI, 7 May 2013. Web. 28 Aug. 2014.

Savage, Britt. "A Guide to Hash Algorithms." *Global Information Assurance Certification.* SANS Institute, 18 Apr. 2003. Web. 28 Aug. 2014.