Mining attacks

Cryptocurrencies Securities

https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf

Josh
Elsa

Minority pool strategy

- Selfish Mining
- Benefits to exploiting this strategy
- Orphan rates
- Other methods of attack

Strategy

- Minority attack by a private pool
- Pool’s branch length vs current blockchain height
  - Lower mining capacity as a pool will lead the pool to time the attack accordingly

Model

- We can define the set of all miners as 1,2,...n miners
- Each miner has computation power m_i
  - The total mining power of a pool is the summation, and we will denote that the sum will be represented as α: $\sum_{i=1}^{n} m_i = 1$
- For the honest miners/ others we can denote their mining power as 1 - α
- Another term used was γ and this is to represent the proportion of miners working on the new created pool's branch and with that 1-γ will be the miners still continuing their work on their initial branch

Concerns involved

- Requirements for a selfish mining pool attack to be implemented

Private Mining Threshold

- Varies as a function of massage propagation speed in the network.
- Once met, “Selfish pool’s rewards exceed its share in network mining power”
  - Revenue of the selfish pool rises to a superlinear growth pattern

$$R_{\text{pool}} = \frac{r_{\text{pool}}}{r_{\text{pool}} + r_{\text{others}}} = \cdots = \frac{\alpha(1-\alpha)^2(4\alpha + \gamma(1-2\alpha)) - \alpha^3}{1 - \alpha(1 + (2-\alpha)\alpha)} . \quad (8)$$

Revenue Graph in relationship to γ

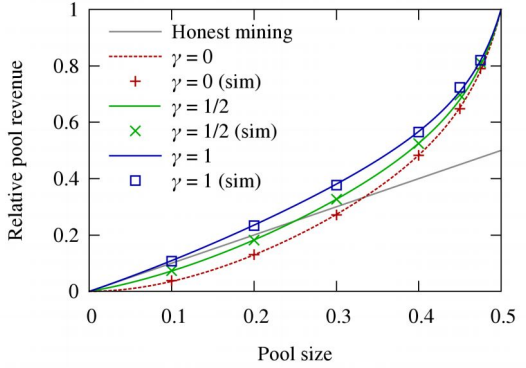


Fig. 2: Pool revenue using the Selfish-Mine strategy for different propagation factors γ, compared to the honest Bitcoin mining protocol. Simulation matches the theoretical analysis, and both show that Selfish-Mine results in higher revenues than the honest protocol above a threshold, which depends on γ.

- Case γ = 1, Case γ = 0.

Minimum Threshold

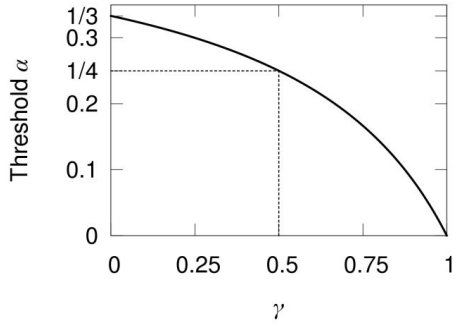


Fig. 3: For a given γ, the threshold α shows the minimum power selfish mining pool that will trump the honest protocol. The current Bitcoin protocol allows $\gamma = 1$, where Selfish-Mine is always superior. Even under unrealistically favorable assumptions, the threshold is never below $1/3$.

Another view from the reading to see the effects of different ratios of miners on the pool's branch, 1 being every miner to 0 with no miners on this forked branch.

Problem and Solution

- Current blockchain protocol of first acceptance and the issue with infiltration of the public's propagation process
- New policy to be adopted by all miners to fix the distribution of miners after two branches of same length have been announced.

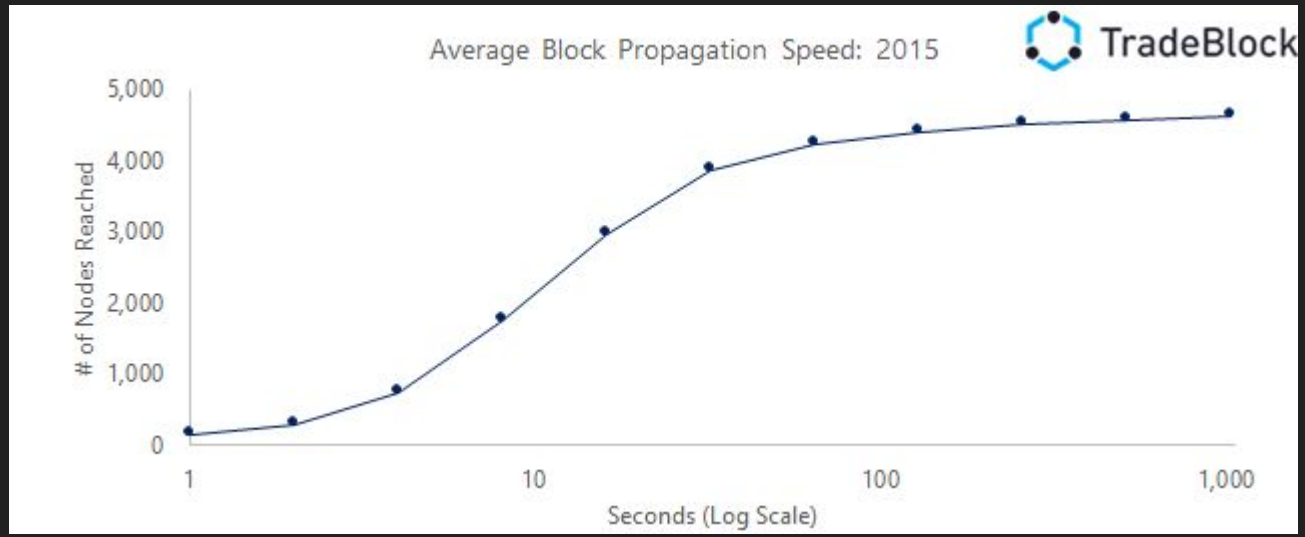
Prevention

https://bitcointalk.org/index.php?topic=441465.msg7282674#msg7282674

Data Propagation

- Referenced Tradeblock’s analysis on this topic
- The dataset comes from a study performed in 2015

Propagation speed:



Orphan rates

When high levels of orphans occur in a day, it can cause suspicion. If the amount does not relate to the average orphan rate of the data set, it is unlikely to be a natural occurrence of two honest miners publishing simultaneously.

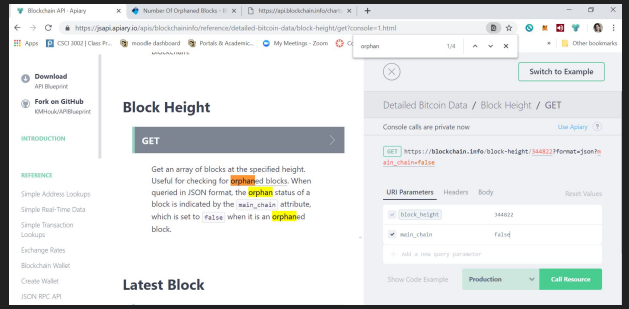
Finding Orphan rates

3 indicators of selfish mining:

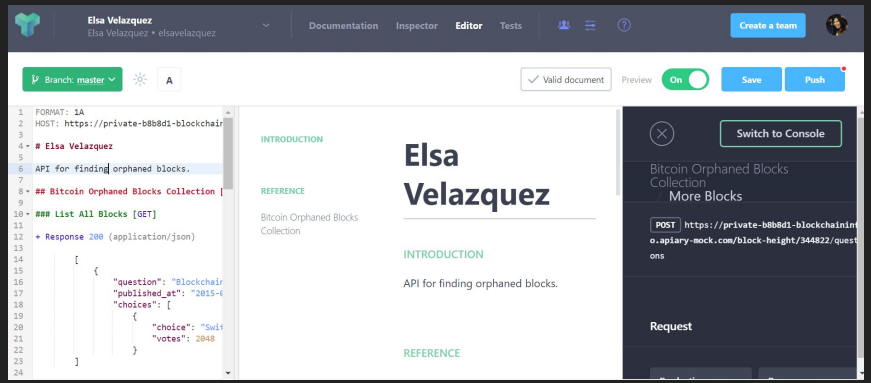
- **Propogation**
- More orphan blocks than expected
- Timing

Step 1- create the API request using jsapi's free (though limited) API

https://jsapi.apiary.io/apis/blockchaininfo/reference/simple-address-lookups/get?console=1.html



https://**blockchain.info**/block-height/344822?format=json?

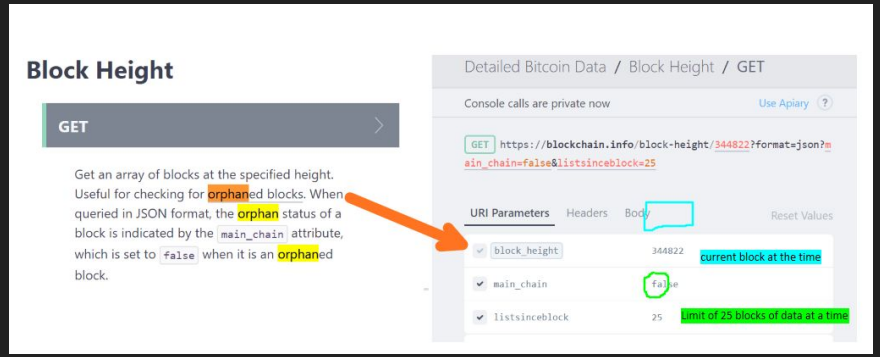


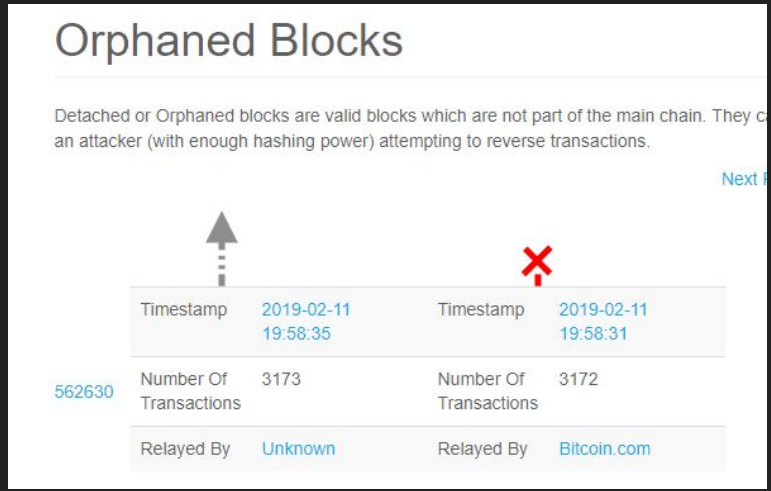
Step 2- if you don’t have a private API, use their public one: https://www.blockchain.com/btc/

Splice

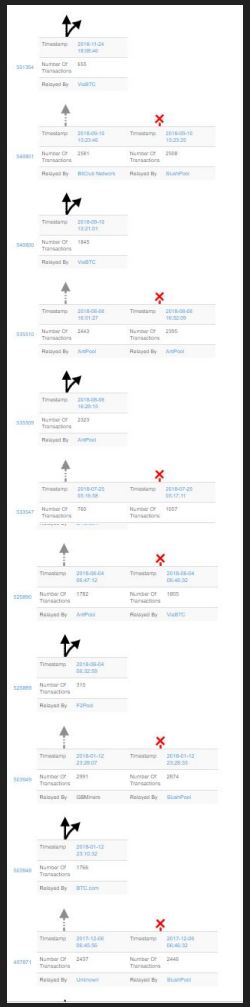
the request format from their API page
https://jsapi.apiary.io/apis/blockchaininfo/reference/json-rpc-api/get.html

their public API address
https://www.blockchain.com/btc/block-height/562630





https://www.blockchain.com/btc/orphaned-blocks



ORPHAN BLOCKS ON THE MAINCHAIN

2019-02-11 Bitcoin.com

2018-11-24 - Antpool

2018-09-10 Slush Pool

2018-08-06 AntPool

2018-07-25 BTC.com

2018-06-04 VIABTC

2018-01-12 Slush Pool

2017-12-06 Slush Pool

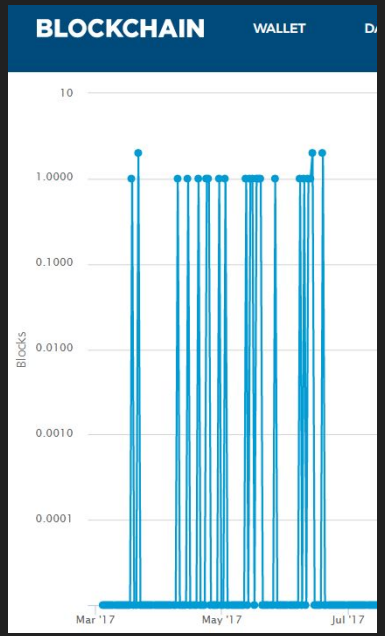
2017-11-20 BTC.TOP

2017-11-16 BTC.com

There are really very few orphan blocks.

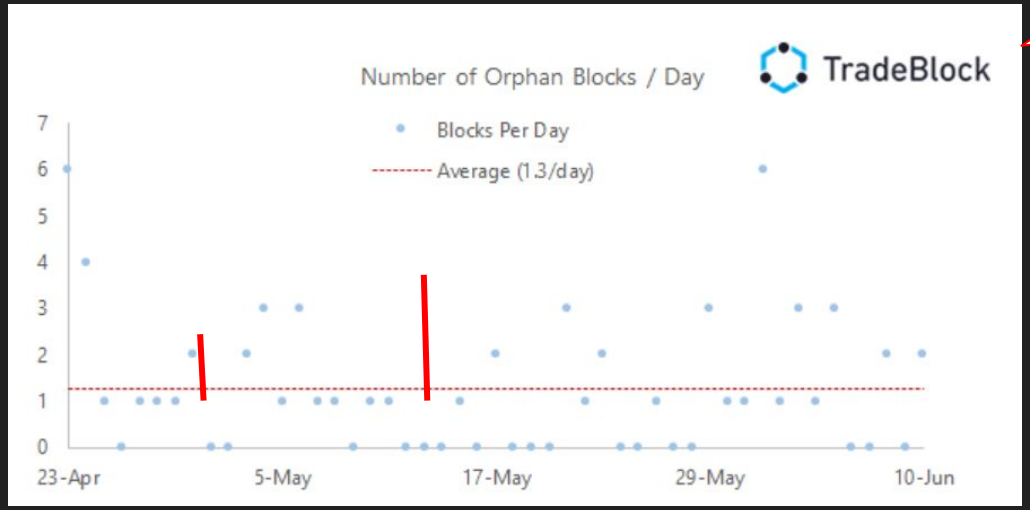
Blockchain.com provides CSV and json files of orphan block data between March 2014 and June 2017.

Orphaned blocks per 24 hours
https://www.blockchain.com/charts/n-orphaned-blocks?timespan=2years&showDataPoints=true



A Selfish Mining Attack Would Look Like This:

https://tradeblock.com/blog/bitcoin-network-capacity-analysis-part-6-data-propagation



"...one should not expect full 8MB blocks in the near-term" - June 2015

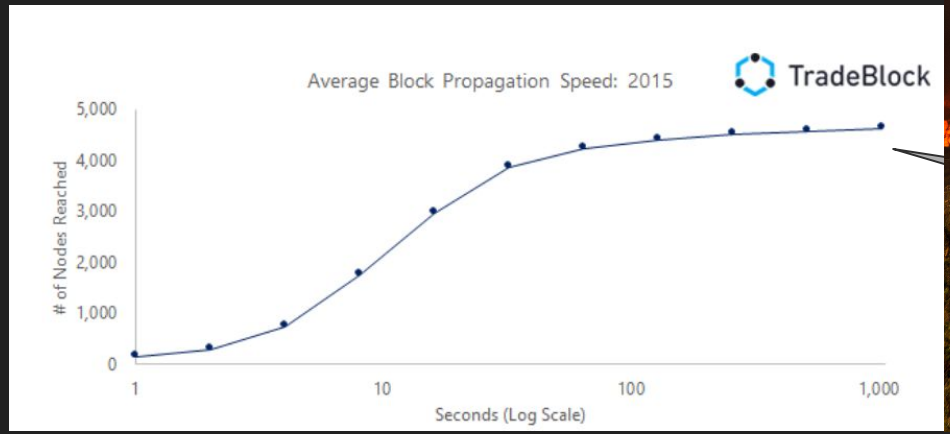
Data Propagation

The data collected by the above histograms is neither recorded nor able to be captured by the Blockchain itself because of Blockchain architecture and because of data propagation.

Mainchain Blockchain only records the competing blocks that made it to, or were propagated to, the most nodes fast enough, and, therefore any competing blocks that didn't make it through the network on time would only be known to the nodes they did reach. They r

Tradeblock put out nodes that were able to collect even c

s able to reach unique insights beyond what is attainable b



Orphan Block Data With Propagation Factor

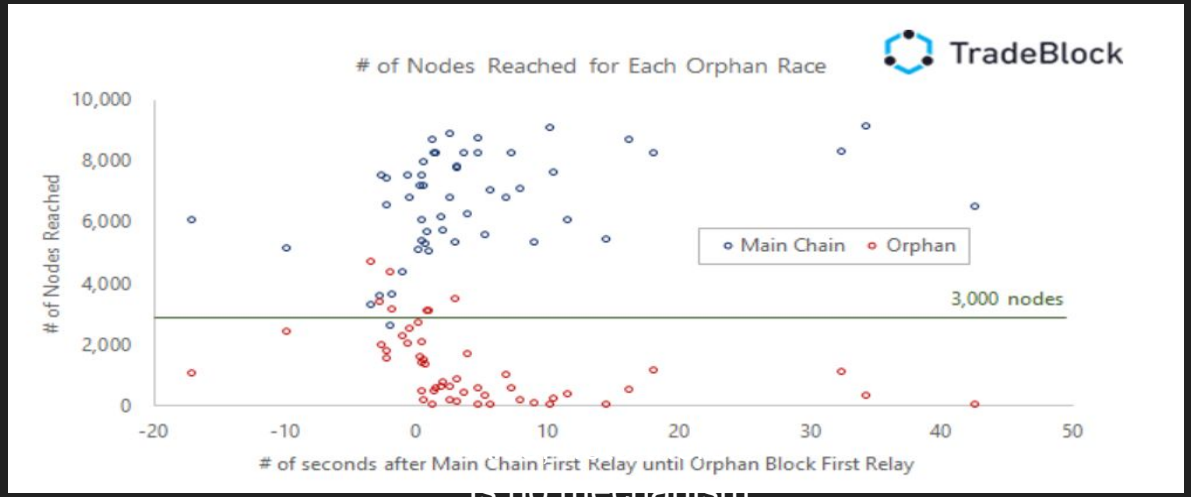
1 indication of selfish mining

The red dots never made it onto the Blockchain, but TradeBlock was able to capture the attempt and how it fared.

.

Therefore, this shows that the one who gets there first and connects to the most nodes fastest, *IS* indeed the winner.

***Note: Most of the red dots are beneath the 3K nodes reached line, while all but 1 **blue dot** are above the line.*



is no mechanism

What this means to the blockchain ecosystem...

Tradeblock reports:

“(1) there is a direct relationship between the size of the block and the time taken to propagate through the network
(2) blocks involved in an orphan race are significantly larger, on average, than blocks that are not in a race
(3) geographic location/ proximity to networks, did not have a notable effect

Therefore miners would not want to publish blocks with more transactions...

However, fortunately...

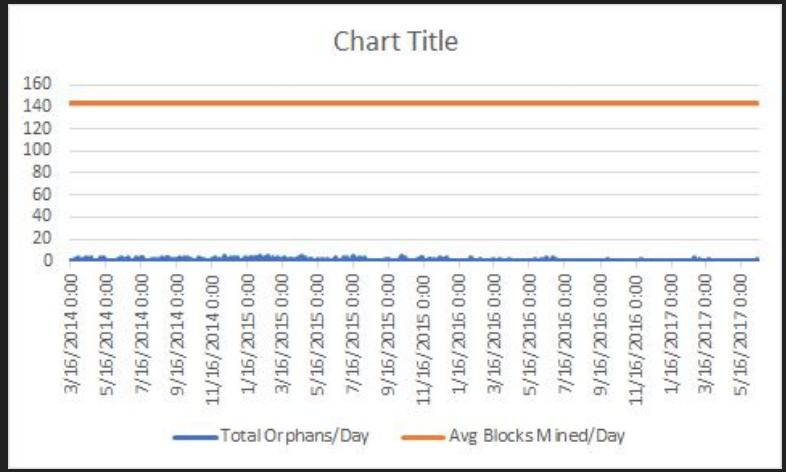
Orphan Block Rate

another indication of selfish mining

We see the orphan blocks rate as compared to the predicted or standard orphan blocks rate

the CSV Data x avg of 144 blocks mined per day =

	Date	Total Orphans
1062	10/24/2014 0:00	4
1063	10/26/2014 0:00	2
1064	10/28/2014 0:00	2
1065	10/30/2014 0:00	1
1066	11/1/2014 0:00	2
1067	11/3/2014 0:00	0
1068	11/5/2014 0:00	1
1069	11/7/2014 0:00	1
1070	11/9/2014 0:00	0
1071	11/11/2014 0:00	0
1072	11/13/2014 0:00	1
1073	11/15/2014 0:00	2
1074	11/17/2014 0:00	0
1075	11/19/2014 0:00	0
1076	11/21/2014 0:00	3
1077	11/23/2014 0:00	2
1078	11/25/2014 0:00	0
1079	11/27/2014 0:00	2
1080	11/29/2014 0:00	2
1081	12/1/2014 0:00	0



Therefore, there does not exist evidence of selfish mining in the Bitcoin blockchain.

However, we don't have Tradeblock's current orphan races data so can't *truly* compare apples to oranges.

The timing between the published block and the orphan block

Are there other attacks on the Bitcoin blockchain?

Asic Boost- dedicated mining hardware

https://arxiv.ftporg//arxiv/papers/1604/1604.00575.pdf

gate count reduction on the silicon

- Increases speed by 20%
- Reduces cost by 20%

Therefore large mining farms stand to profit from using it

This is how it's done without the ASIC Booster

Bitcoin's mining function processes block headers with a double SHA

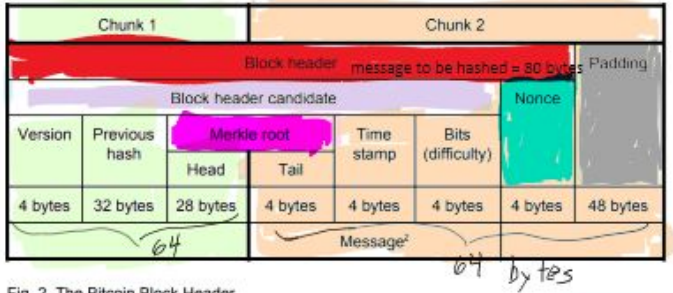


Fig. 2. The Bitcoin Block Header

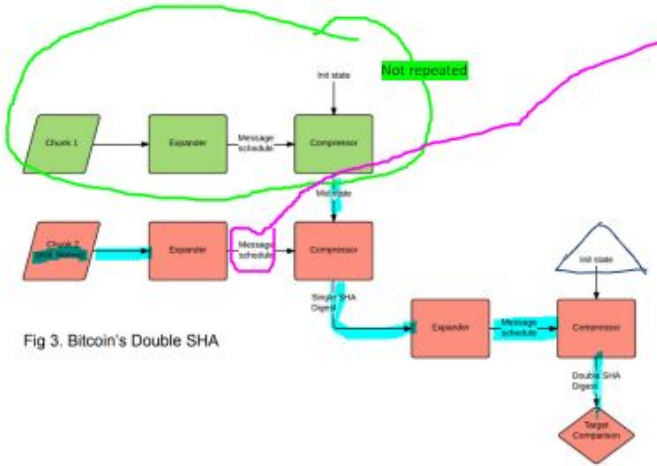


Fig 3. Bitcoin's Double SHA

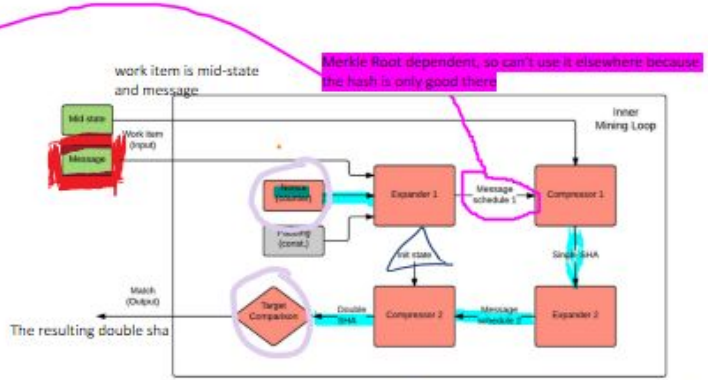


Fig. 4. Bitcoin's Mining Loop

init gets passed in

This is what the ASIC Booster does

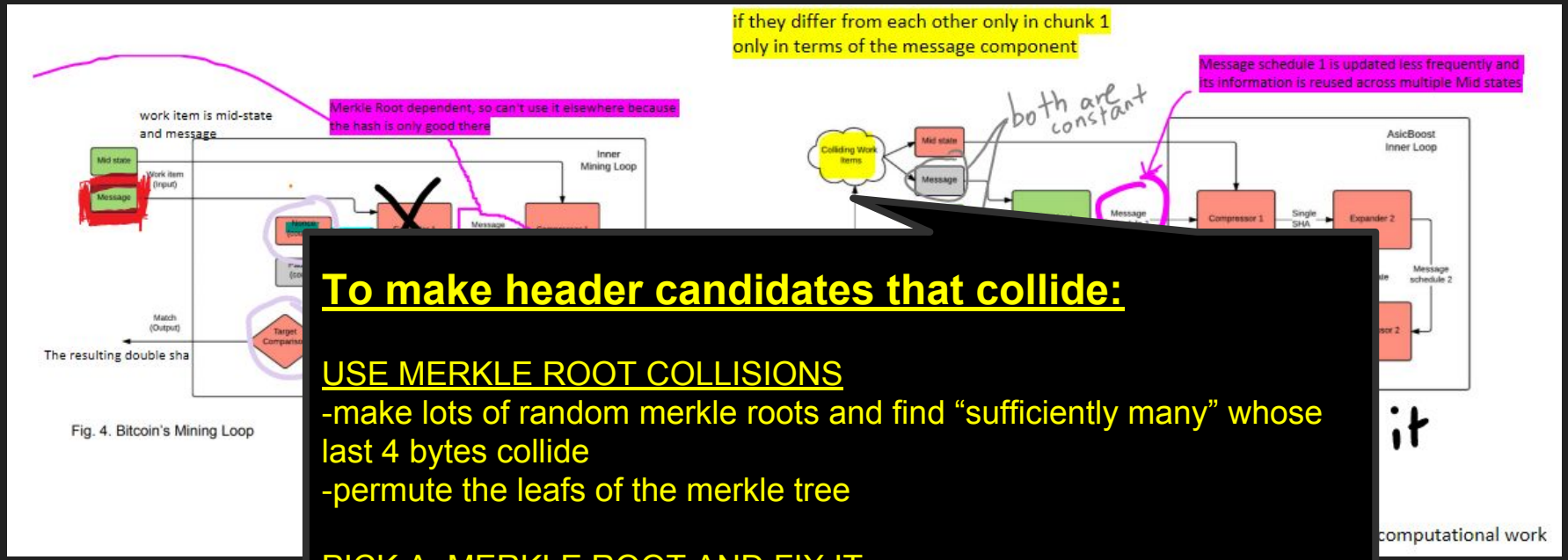


Fig. 4. Bitcoin's Mining Loop

<u>To make header candidates that collide:</u>

<u>USE MERKLE ROOT COLLISIONS</u>
-make lots of random merkle roots and find "sufficiently many" whose last 4 bytes collide
-permute the leafs of the merkle tree

<u>PICK A MERKLE ROOT AND FIX IT</u>
-miner uses bits inside Chunk 1 that aren't in the merkle root
This is extremely efficient because each Mid State gives a new colliding work item (where the last 4 bytes match)

Fig. 6. AsicBoost Gain Percentage

Could there *be* more gains?

Yes!

Spread the gains further by applying the same method to more cores that share Expander 1 between >=2 hashing cores

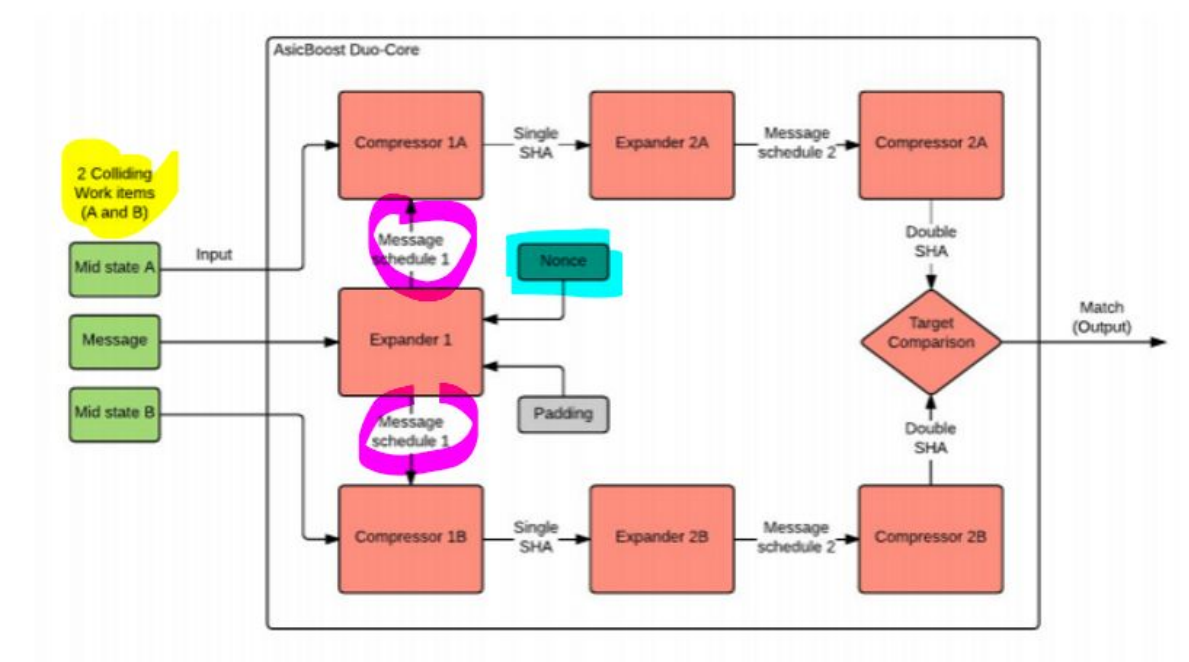


Fig. 7. AsicBoost Duo-core

Are there other ways to game bitcoin blockchain mining?

Yes!

Pool hopping is an issue in some bitcoin mining pools.