Selfish Mining Pools

Analysis of AntPool and F2Pool

Isaac Vawter, Alexey Shablygin
College of Information and Computer Science
University of Massachusetts Amherst
Amherst, MA
ivawter@umass.edu, ashablygin@umass.edu

***Abstract*—In this paper we will analyze blockchain data from two of the biggest mining pools, AntPool and F2Pool, to identify mining activity that is indicative of selfish mining. We look for indication of selfish mining by comparing the theoretical probability and observed probability of mining *n* blocks sequentially given their respective mining power. In addition, we conduct a Monte Carlo simulation to determine the likelihood that sequential block discovery by each miner occurred as expected given their mining power.**

I. Introduction

The objective of this paper is to observe if the frequency of sequential block discovery by individual bitcoin miners is indicative of honest or selfish mining. Selfish mining occurs when a miner uses an algorithm[1] that allows them to take advantage of bitcoin's proof of work system to mine multiple blocks in a row, by mining blocks in secret and only releasing them at the opportune time. This causes competing miners to waste their mining power by not mining from the most recent block, because that block is kept hidden by the selfish miner. Since the selfish mining algorithm[1] requires miners to publish sequences of blocks all in a row, we aim to detect it by analyzing the frequency of block discovery sequences of different lengths by individual miners.

II. Previous Work

This paper makes use of "***Majority Is Not Enough Bitcoin Mining Is Vulnerable***"[1], particularly their definition of selfish mining, the algorithm to perform selfish mining, and findings on probability of selfish mining given computing power of a miner.

III. Methods

In order to conduct our analysis do this we collected data about all of the blocks mined between December 1st, 2015 and April 20th, 2016, including information about which of those blocks were mined by AntPool and F2Pool. We collected our data using three main sources – blockchain.info, antpool.com and f2pool.com. We scrapped blockchain.info to get all of the mined blocks for our time period. Antpool.com and f2pool.com both provided us with data about the blocks each pool mined respectively. We assume that both pools are honest and reported every mined block on their respective websites. After we collected all of block data, we developed a Python program to analyze the data and extract useful statistics on the gathered data. When conducting our analysis, we partitioning blocks into different sample sizes (2000, 1000, 500, 100 blocks) in order to account for fluctuations in mining power and the chance that miners may not be mining selfishly all the time.

Initially we compared the observed probability **P** against the theoretical probability **P*** of sequentially discovering **n** blocks in a row based on the sample size **y**, the number of blocks mined by the mining pool **x,** and the total number of block sequences **z** of length **n** mined by the pool within the sample for each mining pool using the following formulas:

$$\mathbf{P = (x/y)^n}$$

$$\mathbf{P^* = z / (y - (n-1))}$$

While this method did identify some major deviations of **P*** from **P**, the fact that **P*** was consistently lower than **P** seemed to indicate that our equations and/or methods were flawed and so we adopted a different approach. By performing a Monte Carlo simulation on each sample, we were able to determine expected number and standard deviation of sequential block discoveries of varying lengths based mining power of each pool. This was done by repeatedly shuffling an array **x** 1's and (**y** – **x**) 0's, and then counting the total number of sequences of 1's with lengths 1 through 10. After shuffling and recounting 1000 times, the data was output to a file, which was then further analyzed by an R script to create a graphical representation and identify outliers that were more than two standard deviations from what was expected.

IV. Results

Our findings indicate that there were several occurrences of sequential block discovery that were highly suspect within the time period of December 1st, 2015 and April 20th, 2016. During our initial analysis, we identified two separate occurrences where AntPool mined 7 blocks in a row, which is incredibly unlikely given the theoretical probability of such an occurrence being ~ 0.0061%. The following and table shows some of the findings indicating a strong deviation from expected mining behavior and possible indications of selfish mining:

Mining Pool	Sample Start Height	Sample End Height	Sequence Length	Observed Sequences	Expected Sequences	Standard Deviation	Deviations From Expected
F2Pool	407056	406057	3	13	7.867	2.551727	2.011579
AntPool	404056	403057	7	1	0.08	0.274955	3.346008
AntPool	403056	402057	2	26	38.86	5.013821	2.56491
AntPool	402056	401057	2	42	32.281	4.492665	2.163304
F2Pool	400056	399057	3	14	7.867	2.551727	2.40347
AntPool	397055	396054	2	43	30.313	4.319147	2.937386
F2Pool	397055	396054	6	1	0.056	0.234231	4.030215
AntPool	394053	393054	1	166	146.228	9.837175	2.009927
AntPool	394053	393054	2	31	41.366	5.177649	2.002067
AntPool	394053	393054	5	3	0.933	0.938355	2.20279
AntPool	392053	391054	7	1	0.07	0.259037	3.590225
F2Pool	387033	386119	1	19	29.843	1.938131	5.594566
F2Pool	387033	386119	2	5	1.012	0.933732	4.271031
F2Pool	387033	386119	3	1	0.043	0.207728	4.606982

V. Conclusion and Future Work

Through this research we found that it is possible to detect deviations from expected mining behavior, and possibly a sound method for detecting selfish mining. One future direction to take with this work would be to analyze the blockchain of a simulated scenario with selfish and honest miners to see if this method. Another future direction would be two extend the analysis back future in history and include more pools. Finally, another interesting expansion on this analysis would be to combine blocks from multiple mining pools to investigate the likelihood of miners colluding to mask combined selfish mining.

References

[1] Eyal, I., & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In Financial Cryptography and Data Security (pp. 436-454). Springer Berlin Heidelberg.

[2] Black, P. E. (2005). Fisher-yates shuffle. Dictionary of algorithms and data structures, 19.