# CS 765 Assignment 2

Adithya Bhaskar (190050005) Danish Angural (190050028) Dev Desai (190020038)

March 18, 2023

## 1 Deriving the MPU formulae for Selfish Mining

As the Selfish Mining paper [ES13] does not give formulae for the MPU values (and gives only the revenue ratios), we derive them here. First, consider the adversary. The MPU is given by the proportion of the time the adversary's mined block makes it into the main chain. The total steady-state probability of the adversary mining a block is $\alpha$. Of these, every time a block is mined in a state other than $0$, a corresponding (though not necessarily the same) block is guaranteed to make it into the main chain. When a block is mined in state $0$, the only scenario of the mined block not making it into the main chain is when we go to $0'$ next, and thereafter the honest nodes mine a block atop their own previous block. The probability of this happening is

$$(\alpha P_0) \times (1-\alpha) \times (1-\gamma)(1-\alpha) = \alpha \times \frac{(1-\gamma)(1-\alpha)^2(1-2\alpha)}{1-4\alpha^2+2\alpha^3}$$

Therefore

$$\mathsf{MPU}_{\mathsf{adv}} = 1 - \frac{(1-\gamma)(1-\alpha)^2(1-2\alpha)}{1-4\alpha^2+2\alpha^3}$$

For the honest nodes, given that they just mined a block, the probability of their block making it to the main chain is $1$ when in state $0$ or $0'$, $\alpha(1-\gamma)$ when in state $1$, and $0$ otherwise. So their MPU (*Note:* this is not the overall MPU as that would include the adversary as well) should be given by

$$P_0 + P_{0'} + \alpha(1-\gamma)P_1$$

or,

$$\mathsf{MPU}_{\mathsf{others}} = \frac{(1+\alpha-\alpha^2)(1-2\alpha)}{1-4\alpha^2+2\alpha^3} + (1-\gamma)\cdot\frac{\alpha^2-2\alpha^3}{1-4\alpha^2+2\alpha^3}$$

The overall MPU is then

$$\mathsf{MPU}_{\mathsf{overall}} = \alpha\mathsf{MPU}_{\mathsf{adv}} + (1-\alpha)\mathsf{MPU}_{\mathsf{others}}$$

as a fraction $\alpha$ of the generated blocks are from the attacker. We would like to make two comments about this quantity, however. This calculation does not take into consideration the effort gone into trying to mine a block before finding out that a new one has been mined by someone else, and the weight in calculating $\mathsf{MPU}_{\mathsf{overall}}$ is not exactly $\alpha$. This effect is more severe on the honest nodes, as they observe it more. Therefore, the actual value of the overall MPU should be slightly less than the theoretical value in practice. Second, since some of the honest nodes are slow but the attacker is fast, the fraction $\zeta$ of the nodes that the attacker is connected to is likely a slight underestimate of $\gamma$.

# 2 Experimental Results for Selfish Mining

## 2.1 Settings for the experiments

We use the settings suggested in the Problem Statement with one change. As seen above, the transaction generation frequency affects which transactions make it to blocks, but not the mining frequency or reward dynamics. With the default settings of $10$ ms, we observed that our laptop memory (RAM) was exhausted (as we simulate all settings for $100000$ seconds) by the simulation, causing disk swaps to slow down the simulation. Since fewer transactions also mean lesser simulated messages and therefore a faster simulation (although as in P1 the simulated latencies are not affected by those of other messages on the link), we settled on using $45$ s as the default mean transaction generation interval per node. Our "default" settings consist of $n = 100$, $T_k = 10$ min, $T_{tx} = 45$ s, and the fraction of both slow and low-CPU honest nodes being $0.5$. Within this, we simulate the combinations of adversary mining power $P_{adv} \in \{30\%, 40\%\}$ and $\zeta \in \{25\%, 50\%, 75\%\}$. In addition, we also simulate one run each with $25$ nodes, with mean transaction mining time $T_{tx} = 90$ s and one with mining interval mean $T_k = 20$ min to observe the effects (if any) of these ablations. We always assign the attacker the node ID $0$ for easy bookkeeping.

Our metrics include the MPU values $\text{MPU}_{\text{adv}}$ and $\text{MPU}_{\text{overall}}$, as well as the revenue ratio $r_{\text{adv}}$ for the adversary. **Note that for any quantity $x$, we denote by $x$ our measured value and by $\hat{x}$ the theoretical value.**

## 2.2 Experimental Results and Commentary

Our observations for the Selfish Mining case are given in Table 1.

| Defining Parameters | $\text{MPU}_{\text{adv}}$ | $\hat{\text{MPU}}_{\text{adv}}$ | $\text{MPU}_{\text{overall}}$ | $\hat{\text{MPU}}_{\text{overall}}$ | $r_{\text{adv}}$ | $\hat{r}_{\text{adv}}$ |
|---|---|---|---|---|---|---|
| $P_{\text{adv}} = 30\%, \zeta = 25\%$ | 0.846 | 0.788 | 0.783 | 0.752 | 0.338 | 0.300 |
| $P_{\text{adv}} = 30\%, \zeta = 50\%$ | 0.892 | 0.859 | 0.769 | 0.764 | 0.365 | 0.327 |
| $P_{\text{adv}} = 30\%, \zeta = 75\%$ | 0.909 | 0.929 | 0.767 | 0.776 | 0.379 | 0.354 |
| $P_{\text{adv}} = 40\%, \zeta = 25\%$ | 0.897 | 0.889 | 0.720 | 0.690 | 0.504 | 0.505 |
| $P_{\text{adv}} = 40\%, \zeta = 50\%$ | 0.921 | 0.926 | 0.681 | 0.695 | 0.547 | 0.526 |
| $P_{\text{adv}} = 40\%, \zeta = 75\%$ | 0.921 | 0.963 | 0.698 | 0.700 | 0.557 | 0.547 |
| $n = 25, P_{\text{adv}} = 40\%, \zeta = 50\%$ | 0.873 | 0.926 | 0.710 | 0.695 | 0.496 | 0.526 |
| $T_{tx} = 90$ s, $P_{\text{adv}} = 40\%, \zeta = 50\%$ | 0.958 | 0.926 | 0.710 | 0.695 | 0.567 | 0.526 |
| $T_k = 20$ min, $P_{\text{adv}} = 40\%, \zeta = 50\%$ | 0.940 | 0.926 | 0.704 | 0.695 | 0.544 | 0.526 |

Table 1: Experimental results for the selfish mining case.

We note that our values agree with the theoretical numbers, and quite closely in most cases. The slight differences are due to the stochasticity of the runs - since 100000 seconds of simulation corresponds to only $\sim 150$ blocks, randomness plays a role. We make the following observations.

- The overall MPU *increases* as the proportion of nodes connected to the attacker goes up – this is because the increased MPU of the attacker offsets the decrease in the MPU of the other nodes.
- The gains are bigger for $P_{\text{adv}} = 40\%$ as compared to the $30\%$ case, as the theoretical numbers also dictate. One way of looking at this is as follows: the states $2, 3, \cdots$ are *safe* for the attacker in that any block mined

by others will necessarily be orphaned. Therefore, his gains can be higher with less time he spends in states $0, 0'$ and $1$, which increases with his mining power.

- The MPUs and $r$-values are not affected by $n$, $T_{tx}$ or $T_k$ much. In the first and last cases, changes can influence the output variance, e.g. a larger interval between blocks translates to fewer blocks and a more random result behavior. However, they do not influence the means of the relevant values.
- Increasing the fraction of neighbors of the adversary (i.e., $\gamma$) helps less for the $40\%$ mining power case as compared to the $30\%$ one. We can also see in the graph of [ES13] that the three curves corresponding to $\gamma \in \{0.25, 0.50, 0.75\}$ get closer with increased adversary mining power.

## 2.3 Example Blocktrees for Selfish Mining

Our convention is to represent, for a given node $n$'s blocktree, the blocks mined by $n$ in purple if on the main chain and red otherwise. For blocks mined by other nodes, the corresponding colors are green and orange, respectively. The genesis block is displayed in gray.

(a) An example blocktree at the attacker

(b) Corresponding blocktree at an honest node

Figure 1: Example blocktrees for the selfish mining case. The blocktree's image has also been included in the submission where it can be zoomed into.

Example blocktrees at an attacker and an honest node are shown in Figure 1.

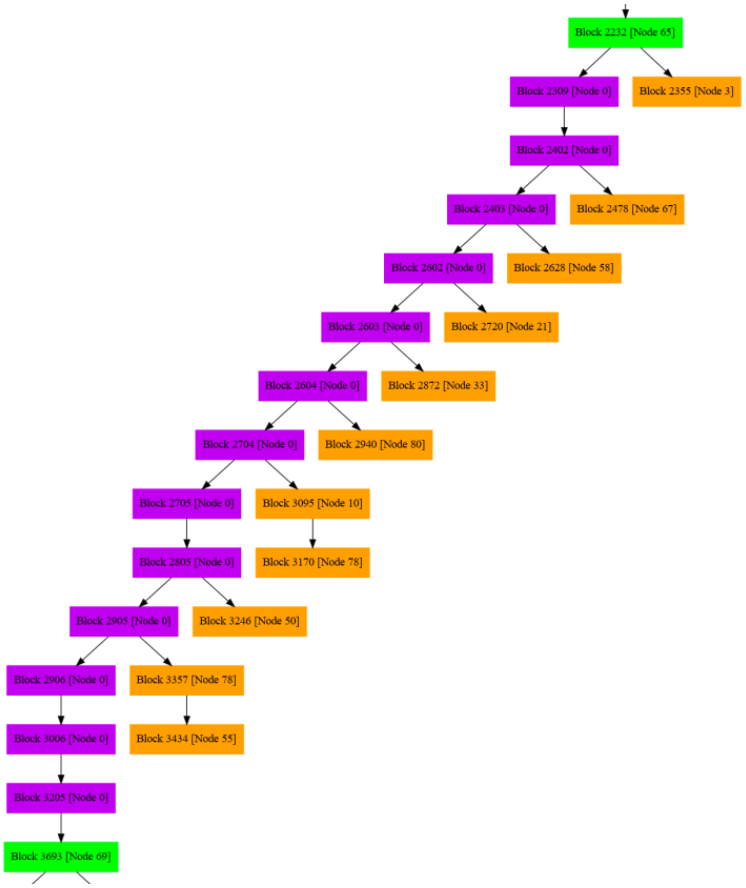
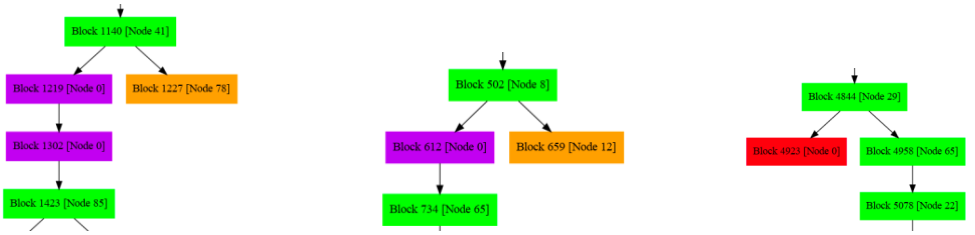



Figure 2: An example of the attacker in a high numbered state and releasing one block every time the others mine one, in the selfish case.

We also show below some zoomed-in snippets of the above blocktree (at the attacker) highlighting his strategy. First, we show in Figure 2 a case where the attacker makes to a high numbered state, and releases one block every time the honest nodes catch up one block as corroborated by the orphaned honest blocks. Finally, when the honest nodes reduce a lead from 2 to 1, the attacker releases both blocks.




(a) Attackers mine a block on top of theirs.

(b) Honest nodes mine a block on top of the attacker's.

(c) Honest nodes mine a block on top of theirs.

Figure 3: The three possible resolutions of state $0'$ for selfish mining.

Figure 3 shows the three possible ways to proceed from a $0'$ state. Either the attacker mines a block atop her previous one, or the honest nodes mine one on top of it, or the honest nodes mine a block on top of their previous block.

# 3 Experimental Results for Stubborn Mining

## 3.1 Settings for the experiments

We use the same settings for Stubborn Mining [Nay+16] as for the Selfish case. However, as we do not have corresponding formulae for the theoretical values of the MPUs and reward ratios, we only report the empirical measurements.

Our metrics include the MPU values $\text{MPU}_{\text{adv}}$ and $\text{MPU}_{\text{overall}}$, as well as the revenue ratio $r_{\text{adv}}$ for the adversary.

## 3.2 Experimental Results and Commentary

Our observations for the Stubborn Mining case are given in Table 2.

| Defining Parameters | $\text{MPU}_{\text{adv}}$ | $\text{MPU}_{\text{overall}}$ | $r_{\text{adv}}$ |
|---|---|---|---|
| $P_{\text{adv}} = 30\%, \zeta = 25\%$ | 0.683 | 0.765 | 0.215 |
| $P_{\text{adv}} = 30\%, \zeta = 50\%$ | 0.868 | 0.750 | 0.289 |
| $P_{\text{adv}} = 30\%, \zeta = 75\%$ | 0.875 | 0.735 | 0.316 |
| $P_{\text{adv}} = 40\%, \zeta = 25\%$ | 0.847 | 0.584 | 0.649 |
| $P_{\text{adv}} = 40\%, \zeta = 50\%$ | 0.923 | 0.603 | 0.638 |
| $P_{\text{adv}} = 40\%, \zeta = 75\%$ | 0.892 | 0.585 | 0.641 |
| $n = 25, P_{\text{adv}} = 40\%, \zeta = 50\%$ | 0.867 | 0.574 | 0.644 |
| $T_{tx} = 90$ s, $P_{\text{adv}} = 40\%$ | 0.974 | 0.621 | 0.644 |
| $T_k = 20$ min, $P_{\text{adv}} = 40\%$ | 0.897 | 0.602 | 0.625 |

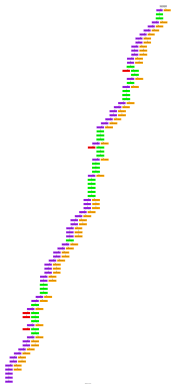
Table 2: Experimental results for the stubborn mining case.

We make the following observations about the observed numbers.

- The gains for the attacker are lesser than selfish mining when her mining power is $30\%$, but greater when it is $40\%$. In other words, stubborn mining pays off more when the attacker controls more of the network. This makes sense, as her strategy is riskier, and loses big whenever the honest chain catches up.
- The fraction of nodes connected to the attacker varies directly with the performance for the $30\%$ mining power case, but the $40\%$ case is not affected much by it.
- As in the selfish mining case, changing $n$, $T_{tx}$ or $T_k$ does not affect the efficacy of the attacker's strategy, in expectation.

### 3.3 Example Blocktrees for Stubborn Mining

Our convention once more is to represent, for a given node $n$'s blocktree, the blocks mined by $n$ in purple if on the main chain and red otherwise. For blocks mined by other nodes, the corresponding colors are green and orange, respectively. The genesis block is displayed in gray.

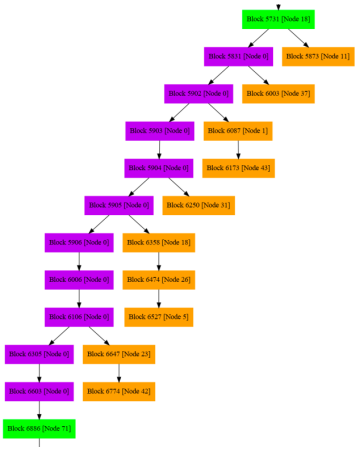


(a) An example blocktree at the attacker

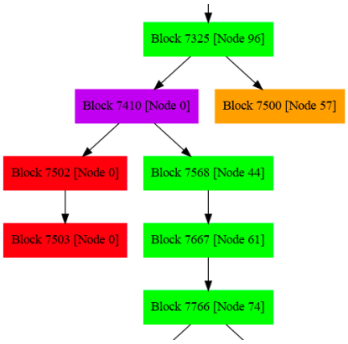
(b) Corresponding blocktree at an honest node

Figure 4: Example blocktrees for the stubborn mining case. The blocktree's image has also been included in the submission where it can be zoomed into.

Example blocktrees at an attacker and an honest node are shown in Figure 4.



(a) The attacker does not rush to reveal her blocks when the honest nodes mine a block in the "2" state. Even in the $0'$ state, the attacker secretly mines on her own chain without revealing it.




(b) An example where stubbornness backfires.

Figure 5: Snippets highlighting the difference of stubborn mining to selfish mining.

We also depict in Figure 5 the ways in which a stubborn miner differs from a selfish miner. First, she does not rush to reveal her two blocks in the "2" state when the honest nodes catch up one more block. Therefore, in the first figure we see that the honest nodes caught up all the way, but in the ensuing tussle, the attacker's chain won out. We can also infer the behavior in the $0'$ state from the same figure, since an orphaned honest chain of length $3$ implies that the attacker transitioned to state $0'$ then mined a block on top of her own chain but kept it secret (a chain of length $2$ would be possible in the selfish case due to how states $0'$ and $1$ are handled, but $3$ should be rare). The second image shows an example of the attacker's stubbornness backfiring: when she was two blocks ahead and the honest nodes caught up one block, she released only one block. The miners mining on the former won out to level the scene, and then another block was mined on the honest chain, orphaning both of the attacker's blocks.

# 4 Submission Structure

As the logs take up a combined space of more than $8.6$ GBs, we do not include them in our submission. Instead, we make them available on Google Drive[1]. Our submission includes the report in `report.pdf`. Instructions to reproduce our results are given in `README.md`. We also include the images of this report to be viewed in their full size under `report-images/`.

[1] https://drive.google.com/file/d/167krps9rqVCWx7jiDeuj2a2unkqnjlRc/view?usp=sharing

# References

[ES13] Ittay Eyal and Emin Gun Sirer. *Majority is not Enough: Bitcoin Mining is Vulnerable*. 2013. DOI: 10.48550/ARXIV.1311.0243. URL: https://arxiv.org/abs/1311.0243.

[Nay+16] Kartik Nayak et al. "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2016, pp. 305–320. DOI: 10.1109/EuroSP.2016.32.