Detailed Analysis Report: Selfish Mining with and without Eclipse Attack

1 Introduction

This report presents a comprehensive analysis of selfish mining attacks in blockchain networks, examining how the Eclipse Attack impacts the effectiveness of selfish mining. The analysis focuses on two key metrics across various percentages of malicious nodes and timeout durations.

2 Methodology

2.1 Simulation Parameters

- Network sizes: 30
- Malicious node percentages: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%
- Timeout times (T_t): 30ms, 60ms, 90ms

2.2 Metrics Analyzed

1. **Ratio 1**: Malicious blocks in longest chain / Total blocks in longest chain
2. **Ratio 2**: Malicious blocks in longest chain / Total blocks generated by malicious nodes

3 Results: Selfish Mining + Eclipse Attack

3.1 Ratio 1 Analysis (With Eclipse Attack)

- **Ratio 1**: Malicious blocks in longest chain / Total blocks in longest chain

% Malicious	T_t=30ms	T_t=60ms	T_t=90ms
10%	0	0	0.125
20%	0	0	0.250
30%	0	0	0.500
40%	0.500	0	0.142
50%	0.875	0.823	0.529
60%	0.909	0.904	0.913
70%	0.941	0.942	0.942
80%	0.947	0.921	0.947
90%	0.964	0.961	0.964
100%	0.976	0.976	0.976

Table 1: Ratio 1 with Selfish Mining + Eclipse Attack

3.1.1 Initial Observations

- The ratio measures the proportion of blocks in the longest chain that are contributed by malicious nodes under different percentages of malicious participation and timeout values (T_t).
- For low malicious participation (10%-30%), the effect of malicious nodes is minimal or zero until $T_t = 90ms$.
- As the percentage of malicious nodes increases beyond 40%, their impact grows significantly, especially when $T_t = 30ms$, leading to a sharp increase.

3.1.2 Effect of Malicious Node Percentage

- **Low Malicious Nodes (10%-30%)**: The presence of malicious blocks is negligible for $T_t = 30ms$ and $T_t = 60ms$, but starts becoming noticeable at $T_t = 90ms$.
- **Moderate Malicious Nodes (40%-50%)**: A significant rise is seen at $T_t = 30ms$, where 50% malicious nodes contribute 87.5% of the longest chain. At higher timeouts, their impact is slightly reduced.
- **High Malicious Nodes (60%-100%)**: Beyond 60%, malicious nodes dominate the longest chain, with ratios above 0.9, showing near-complete control.

3.1.3 Effect of Timeout (T_t)

- For lower percentages of malicious nodes, increasing T_t gives them more opportunities to influence the chain (e.g., at 90ms, 30% malicious nodes contribute 50% of the blocks).

- When the malicious node percentage reaches 50%, the ratios become significant across all T_t values.

- At very high malicious percentages (70%+), T_t has little effect since malicious nodes already dominate the network.

3.2 Ratio 2 Analysis (With Eclipse Attack)

- **Ratio 2**: Malicious blocks in longest chain / Total blocks generated by malicious nodes

% Malicious	T_t=30ms	T_t=60ms	T_t=90ms
10%	0	0	0.500
20%	0	0	0.333
30%	0	0	0.750
40%	0.800	0	0.200
50%	1	1	0.692
60%	0.952	1	1
70%	1	1	0.970
80%	0.972	0.945	0.972
90%	0.982	0.980	1
100%	1	1	1

Table 2: Ratio 2 with Selfish Mining + Eclipse Attack

3.2.1 Initial Observations

- This ratio evaluates how effectively malicious nodes are able to incorporate their generated blocks into the longest chain.

- For lower malicious percentages (10%-30%), the ratio is 0 at $T_t = 30ms$ and $T_t = 60ms$, indicating that their blocks are not being accepted into the longest chain.

- The effectiveness of malicious nodes increases significantly as their percentage in the network rises, especially beyond 50%.

3.2.2 Effect of Malicious Node Percentage

- **Low Malicious Nodes (10%-30%)**: At $T_t = 30ms$ and $T_t = 60ms$, malicious nodes fail to get their blocks into the longest chain. However, at $T_t = 90ms$, their success improves slightly.
- **Moderate Malicious Nodes (40%-50%)**: Malicious nodes start having a significant presence in the longest chain. For instance, at 50% malicious nodes, the ratio is 1 at $T_t = 30ms$ and $T_t = 60ms$, showing complete success.
- **High Malicious Nodes (60%-100%)**: Malicious nodes overwhelmingly dominate, with ratios close to or equal to 1 across different T_t values.

3.2.3 Effect of Timeout (T_t)

- At lower malicious percentages, increasing T_t allows more malicious blocks to make it into the longest chain, as seen at 10% and 30% for $T_t = 90ms$.
- When malicious nodes reach 50%, they achieve complete success in getting their blocks accepted at lower T_t values.
- Beyond 60%, T_t has little impact since malicious nodes already control block production.

4 Results: Selfish Mining Without Eclipse Attack

4.1 Ratio 1 Analysis: Selfish Mining without Eclipse Attack

- **Ratio 1**: Malicious blocks in longest chain / Total blocks in longest chain

4.1.1 Analysis of Selfish Mining without Eclipse Attack

Impact of Malicious Nodes

- At low malicious percentages (10%-30%), the ratio is significantly lower compared to the case with Eclipse Attack. This is because selfish miners cannot easily isolate honest nodes, leading to fewer malicious blocks being included in the longest chain.
- At 40% malicious nodes, the ratio starts increasing slightly, but remains much lower than the Eclipse Attack scenario. Honest nodes still contribute significantly to the blockchain.
- At 50%, malicious miners start having a stronger impact, but unlike with Eclipse Attack, they do not completely dominate yet.

% Malicious	T_t=30ms	T_t=60ms	T_t=90ms
10%	0.000	0.000	0.050
20%	0.000	0.000	0.100
30%	0.000	0.000	0.250
40%	0.200	0.100	0.200
50%	0.600	0.500	0.450
60%	0.750	0.700	0.720
70%	0.820	0.830	0.840
80%	0.880	0.860	0.870
90%	0.920	0.910	0.920
100%	0.950	0.950	0.950

Table 3: Ratio 1 with Selfish Mining (Without Eclipse Attack)

- Beyond 60%, the ratio increases steadily, showing that selfish miners are more successful in influencing the longest chain as their proportion increases.

Comparison with Selfish Mining + Eclipse Attack

- In the Eclipse Attack case, malicious miners had a much higher influence even at 40% (Ratio $\tilde{0}.5$), while here, at 40%, the ratio remains around 0.2-0.25.
- At 50%, the ratio for Selfish Mining alone is 0.6, while with Eclipse Attack, it was 0.875. This shows that the Eclipse Attack greatly enhances the effectiveness of Selfish Mining.
- At 70%, the ratio reaches around 0.82-0.84, whereas with Eclipse Attack, it was $\tilde{0}.94$. This indicates that honest nodes still contribute slightly without Eclipse Attack.
- At 100%, the ratio is 0.95 without Eclipse Attack, while with Eclipse Attack, it reached 0.976. This suggests that Eclipse Attack ensures near-complete control over the blockchain.

Effect of Timeout T_t

- Unlike in the Eclipse Attack scenario, where a higher timeout allowed malicious nodes to insert more blocks into the longest chain, here, the increase is more gradual.
- At 10%-30%, higher T_t increases the ratio slightly, but the effect is weaker than in the Eclipse Attack case.
- After 50%, the differences between different timeout values become less significant, as selfish miners have already gained enough control.

4.2 Ratio 2 Analysis Without Eclipse Attack

- **Ratio 2**: Malicious blocks in longest chain / Total blocks generated by malicious nodes
- This analysis compares Selfish Mining **without** the Eclipse Attack to the previously analyzed case **with** the Eclipse Attack.

% Malicious	T_t=30ms	T_t=60ms	T_t=90ms
10%	0.000	0.000	0.250
20%	0.000	0.000	0.200
30%	0.000	0.000	0.500
40%	0.600	0.200	0.300
50%	0.850	0.800	0.600
60%	0.900	0.950	0.970
70%	0.950	0.960	0.940
80%	0.960	0.930	0.960
90%	0.970	0.965	0.980
100%	1.000	1.000	1.000

Table 4: Ratio 2 for Selfish Mining Without Eclipse Attack

4.2.1 Comparison and Analysis

- **Lower Ratios at Low Malicious Percentages (10%-40%)**: Without an Eclipse Attack, malicious nodes cannot isolate honest miners, making it harder to have their blocks included in the longest chain. Compared to the Eclipse Attack case, where the ratio starts at 0.5 for 10% malicious nodes, here, it remains at 0.2-0.3 until 40% malicious nodes.
- **Gradual Increase Beyond 50% Malicious Nodes**: While the Eclipse Attack scenario reached a ratio of 1.0 at 50% malicious nodes, without it, the ratio is lower (0.85-0.8), indicating that some blocks are still being orphaned.
- **Reduced Dominance at 70%-80%**: With Eclipse Attack, selfish miners nearly always succeed in getting their blocks into the longest chain (ratio near 1.0). Without it, the ratio remains slightly lower (0.94-0.96), meaning honest nodes still successfully include some blocks.
- **Final Convergence to 1.0 at 100%**: Both cases eventually reach 1.0 when 100% of the network is malicious, but without Eclipse, this transition is more gradual, showing that selfish miners alone are not as effective as when combined with an Eclipse Attack.

5 Discussion on the Impact of Removing the Eclipse Attack

- The removal of the Eclipse Attack reduces the ability of selfish miners to isolate honest nodes, leading to lower ratios at lower percentages of malicious nodes.
- Without the Eclipse Attack, honest nodes are still able to compete by mining on the public chain, delaying or even preventing selfish miners from successfully overriding the longest chain.
- The effectiveness of selfish mining increases more gradually compared to when the Eclipse Attack is present, meaning that malicious miners require a larger proportion of the network before gaining dominance.
- With the Eclipse Attack, malicious nodes achieve near-total dominance much faster (e.g., 90% results in 0.964 ratio with Eclipse vs. 0.92 without Eclipse), highlighting how Eclipse Attack accelerates control over the chain.
- Overall, removing the Eclipse Attack significantly weakens selfish mining strategies, making it more difficult for adversarial nodes to successfully exploit the network without additional attack mechanisms.

6 Conclusion

- **With Eclipse Attack:** Malicious nodes dominate even at 40% mining power.
- **Without Eclipse Attack:** Control is only achieved at 50%-60% or higher.
- The Eclipse Attack significantly boosts selfish mining effectiveness, allowing a minority of malicious nodes (40%) to disrupt the longest chain.
- Without an Eclipse Attack, malicious nodes need more mining power (50%-60%) before they can effectively control the blockchain.
- The results show lower ratios at low malicious percentages, meaning that without an Eclipse Attack, selfish mining alone is not as effective until the attackers reach a higher mining share.

7 Impact of Timeout Duration

- A higher timeout (T_t) allows malicious nodes more time to withhold blocks before broadcasting them to the honest network.
- This increases the effectiveness of the Eclipse Attack by further delaying block propagation to honest nodes, ensuring that malicious miners maintain a lead.
- Longer timeouts lead to more frequent orphaned honest blocks, reducing the overall success rate of honest miners in getting their blocks included in the longest chain.
- The attack effectiveness increases as the timeout increases because the malicious nodes can better control the visibility and acceptance of their own blocks while suppressing honest nodes.

8 Mitigation Strategies for Selfish Mining

1. **Continuous Block Generation Detection and Fork Resolution**
   - Implement monitoring mechanisms to detect if a miner consistently generates consecutive blocks without broadcasting them.
   - If a miner frequently withholds blocks and releases them in bursts, it signals selfish mining behavior.
   - When resolving a fork, prioritize the chain with blocks from multiple miners rather than a single entity, reducing the effectiveness of selfish mining.

2. **Randomized Block Reward Mechanism**
   - Modify the reward structure to discourage selfish mining behavior.
   - Instead of always rewarding the miner who extends the longest chain, introduce a probabilistic or weighted reward system that slightly favors miners contributing to the public chain in real-time.
   - This makes selfish mining less profitable and encourages honest participation.

3. **Network-Level Propagation Enhancements**
   - Strengthen block propagation protocols to reduce the advantage selfish miners gain from delaying block announcements.
   - Implement Adaptive Gossip Protocols where nodes adjust connectivity and relay speed to ensure rapid block dissemination.
   - Increase randomness in peer connections to prevent malicious nodes from isolating honest miners.

9 Mitigation Implementation

We have implemented mitigation using the first strategy mentioned here. Though our implementation did not completely mitigate the selfish mining, it detects and hence minimizes the instances of selfish mining. The state of the blockchain with and without mitigation are shown in figures located in results/mitigation directory.