Abstract

This article analyzes the security ofProof of Random Access (PoRA)consensus mechanism against potential mining attacks. The focus is on two main attack vectors: the shrink attack, where an attacker uses cheaper or smaller storage devices with the same throughput to gain an economic advantage, and the Moore attack, where an attacker leverages advancements in technology to increase throughput while maintaining the same cost. The article examines these attacks under both unlimited and limited mining throughput scenarios and provides mathematical analysis to determine the conditions under which the attacks could be economically viable for miners.

Security model

In our security model, we consider an attacker who aims to optimize their mining procedure to gain an economic advantage over honest miners. The attacker can employ various strategies to achieve this goal:

1. Equipment replacement: The attacker may replace their storage equipment with more efficient or cost-effective alternatives. This strategy allows the attacker to maintain or increase their mining performance while reducing costs.

2. Partial fraction mining: The attacker can utilize a portion of their storage space and throughput for one client while using the remaining resources for another client. This allows the attacker to optimize their resource allocation and potentially gain an advantage by serving multiple clients simultaneously.

We will analyze two specific attack vectors:

1. Shrink attack: In this attack, the attacker replaces their storage device with a cheaper one that offers the same performance. For example, an attacker might replace a more expensive 1TiB NVMe drive with a cheaper 1TB drive that has the same throughput. This allows the attacker to reduce their storage costs while maintaining their mining performance. The key idea behind the shrink attack is that the attacker can take advantage of the fact that the consensus mechanism may not distinguish between storage devices of different sizes as long as they offer the same throughput.

2. Moore attack: This attack involves the attacker replacing their storage equipment with a newer, more advanced version that offers better performance at the same cost. For example, an attacker might replace a gen 4 storage device with a gen 5 device that has twice the throughput at the same cost. This allows the attacker to increase their mining performance without incurring additional expenses.

It is important to note that the Proof of Random Access (PoRA) consensus mechanism is well-scalable, meaning that there is no single machine limitation. This scalability allows for the possibility of using RAM rigs on tiny devices, such as single-board computers or even smartphones, to participate in the mining process. The absence of a single machine limitation opens up new opportunities for attackers to optimize their mining setups and potentially gain an advantage over honest miners.

Furthermore, the ability to perform partial fraction mining, where an attacker can allocate a portion of their storage space and throughput to one client while using the remaining resources for another client, adds another layer of complexity to the security model. This flexibility in resource allocation allows attackers to optimize their mining strategies and potentially gain an edge by serving multiple clients simultaneously.

Throughout the following sections, we will examine these attack vectors in detail, considering both unlimited and limited mining throughput scenarios. Our analysis will focus on determining the conditions under which these attacks could be economically viable for miners, and we will provide mathematical derivations to support our findings

Shrink attack

In the following sections, we consider three different attacks. The first attack assumes that the mining throughput is not limited and shows how in this case the adversary can gain advantage over honest miners by using a different hardware configuration. In the follwing two attacks, we assume that the mining throughput was limited to mitigate the first attack, but we show that new issues arise: adversary can be economically incentivised to drop some part of the stored file, and perform Moore attack using more efficient hardware than that of honest miners.

In our costs analyses we make a few reasonable assumptions about the relationship between the costs and make conservative estimates of adversary advantage.

Unlimited throughput: achieving advantage over honest miners

In this scenario, we consider the case where an attacker reduces the size of their memory module to gain an economic advantage. We make a pessimistic assumption that if the memory size is reduced by half, the maintenance cost (energy consumption) and throughput will remain the same, while the cost will be reduced by half. In reality, the energy consumption would likely be lower, but this assumption can only make our analysis more conservative.

To compare the cost efficiency of the attacker and the reference miner, we normalize the values by the cost and present them in the following table:

	Cost	Maintenance	Throughput
reference	1	A	1
attacker	1	χA	χ

- The reference miner’s cost of purchasing one unit of hardware is set to 1 as a baseline; this is without loss of generality as we normalize other values with respect to this, eliminating a free variable.
- $A \sim 1$

represents the maintenance cost (energy consumption) per time unit for the reference miner, which is assumed to be close to 1 for simplicity.

- The reference miner’s throughput is also normalized to 1.
- The attacker’s cost is set to 1, assuming that one unit of attacker’s hardware has the same cost as that of a reference miner. This is a conservative estimate, since one unit of attacker’s hardware is assumed to be less efficient than that of a reference miner.
- The attacker’s maintenance cost is χA

, where $\chi > 1$

represents the throughput advantage of the attacker. This is because the attacker’s memory size is smaller, but the energy consumption per unit of memory remains the same.

- The attacker’s throughput is χ

, reflecting their advantage in terms of throughput per unit cost.

To compare the total cost efficiency, we calculate the throughput per unit of total cost (cost + maintenance) for both the reference miner and the attacker:

Reference miner: $\frac{1}{1+A}$

Attacker: $\frac{\chi}{1+\chi A} = \frac{1}{1/\chi+A}$

Since $\chi > 1$

when the attacker reduces the memory size, we can conclude that:

$\frac{1}{1+A} < \frac{\chi}{1+\chi A} = \frac{1}{1/\chi+A}$

This inequality demonstrates that the attacker has a better total cost efficiency compared to the reference miner.

Therefore, the original PoRA is not resistant to shrink attacks under unlimited mining throughput conditions. The only way to protect against this vulnerability is to limit the mining rewards, which would discourage attackers from exploiting this weakness.

Limited throughput: not storing part of the file

In this scenario, we consider the case where the mining throughput is limited to an optimal value of 1, and we analyze the cost efficiency for an attacker who uses only a fraction p

of their memory.

Let’s define the following variables:

- n

: the number of random accesses

- $q = 1 - p \ll 1$

, assuming $qn \lesssim 1$

- n_e

: the efficient average number of accesses, given by $n_e = (1-p^n)/(1-p) \approx (1 - \exp(-qn)) / q$

- p_s

: the success probability, given by $p_s = p^n \approx \exp(-qn)$

- τ

: the slowdown of sampling, given by $\tau = n_e/(n \cdot p_s) = (\exp(qn)-1)/(qn)$

- B

: the sampling cost, assumed to be much smaller than 1 ($B \ll 1$

) to ensure that the main cost of the algorithm is not CPU PoW

We can compare the reference miner and the attacker using the following table:

	Cost	Maintenance	Sampling	Throughput
reference	1	A	B	1
attacker	1	χA	χB	χ
attacker	p	$\chi p A$	$\chi p B\tau$	χp

- The reference miner’s costs and throughput are normalized to 1.
- The attacker’s cost and throughput are scaled by χ

when using the full memory.

- When the attacker uses only a fraction p

of their memory, their cost, maintenance, and throughput are scaled by p

, while the sampling cost is scaled by $p\tau$

to account for the slowdown in sampling.

For $qn \lesssim 1$

, we have $\tau \sim 1$

, which means $B\tau \ll 1$

.

To consume all throughput, the attacker must satisfy the equation: $\chi p = 1$

.

For efficient mining, the following condition must be met:

$p\cdot (1 + \chi A) + B \tau < 1 + A + B$

Simplifying this condition, we get:

$p + (\tau - 1) B < 1$

$q > (\tau - 1) B \approx qnB/2$

$n B \lesssim 2$

To estimate B

, let’s consider the example of a Samsung 970 SSD with a throughput of 2GB/s, TDP of 6W, and a value size of 1MB. The hash efficiency for CPU is 30MH/J, and for ASIC, it is 3GH/J.

The additional TDP for sampling will be:

- For CPU: $2e9/1e6/30e6 = 6\text{e-5}W$
- For ASIC: $2e9/1e6/3e9 = 6\text{e-7}W$

By dividing these values by the TDP, we can roughly estimate B

to be in the range of $1\text{e-}5$

to $1\text{e-}7$

.

This means that n

should be greater than $1\text{e}5$

to $1\text{e}7$

to make the shrink attack inefficient, which may not be practical in real-world scenarios.

When $qn \lesssim 1$

, p

can take any value. For example, with a storage size of 1TB, value size of 1MB, $n=1\text{e}4$

, and $B=1\text{e-}5$

, we get $q=1\text{e-}4$

, which means that 100 MB of data could be forgotten while still providing economic benefits for the miner.

Limited throughput: Moore attack

When considering the Moore attack, it’s important to note that miners will align their throughput to the limit imposed by the system. Let’s analyze the cost efficiency of the reference miner and the attacker in this scenario.

Cost + Maintenance

Sampling

Throughput

reference

1

B

1

attacker

1

B

χ

attacker

p

$\theta Bp\tau$

$\theta \chi p$

- The reference miner’s cost, maintenance, and throughput are normalized to 1, with a sampling cost of B

.

- After upgrading their hardware, the attacker’s cost and maintenance remain the same, but their throughput increases by a factor of χ

.

- When the attacker uses only a fraction p

of their upgraded hardware, their sampling cost is scaled by $\theta p\tau$

, where θ

is a throughput utilization parameter, and their throughput is scaled by $\theta \chi p$

.

θ

represents the throughput utilization parameter, which indicates the fraction of the attacker’s upgraded throughput that they actually use. For example, if $\theta = 0.8$

, the attacker is utilizing 80% of their upgraded throughput. This parameter allows us to model situations where the attacker may not be using their full upgraded capacity, either intentionally or due to technical limitations.

To consume all available throughput, the attacker must satisfy the equation: $\theta \chi p = 1$

.

For efficient mining, the following condition must be met:

$p \cdot (1 + \theta B \tau) < 1 + B$

Expanding this condition using the approximations for τ

and p_s

from the previous chapter, we get:

$(1-q) (1 + \chi^{-1} B (1 + qn/2)) < 1 + B$

Simplifying further:

$\chi^{-1} nB/2 - 1 - B - \chi^{-1} qnB/2 < 0$

$\chi^{-1} pnB/2 < 1 + B$

$n B \lesssim 2 \chi$

To find the optimal values for the arbitrary parameters θ

and p

, we need to perform additional calculations. Taking the partial derivative of $p \cdot (1 + \chi^{-1} B \tau)$

with respect to q

, we get:

$\partial_q (p \cdot (1 + \chi^{-1} B \tau)) \approx -(1 + \chi^{-1} B (1 + qn/2) + \chi^{-1} Bn/2 \cdot (1 + qn/3)) = \chi^{-1} Bn/2 - (1 + \chi^{-1} B) + \chi^{-1} B q n (n / 3 - 1/2) = 0$

Solving for q

, we get:

$q = \frac{-Bn/2 + \chi + B}{B n (n / 3 - 1/2)} > 0$

This result suggests that n

should be greater than $1\text{e}5$

to $1\text{e}7$

to make the Moore attack inefficient.

For example, consider a storage size of 1TB, value size of 1MB, $n=1\text{e}4$

, $B=1\text{e-}5$

, and $\chi=2$

. Plugging these values into the equation for q

, we get $q=0.005$

, which means that 5GB of data could be forgotten while still providing economic benefits for the attacker.

RAM rig

In the original PoRA paper, the authors compare the performance of a Samsung 970 EVO NVMe SSD and 256GB DDR4-3200 RAM. Based on the calculations in the previous sections, we arrive at a counterintuitive conclusion: when there are no throughput limitations, only the throughput matters, not the size of the storage. To further illustrate this point, let’s compare the efficiency of a Crucial T705 1TB NVMe SSD and Crucial 8GB DDR5-4800 RAM.

	Cost (USD)	TDP (W)	Throughput (GB/S)
NVMe	188	15	13.7
DDR5			

25

10

72

The table above compares the cost, thermal design power (TDP), and throughput of the two storage devices. The NVMe SSD has a higher cost and TDP but a lower throughput compared to the DDR5 RAM.

To calculate the cost efficiency of each device, we need to consider the maintenance cost and the amortization of the equipment over its lifetime. Let's assume that the maintenance cost for 1W of power is about 4.4 USD per year and that the equipment is amortized over 4 years.

For the NVMe SSD, the cost per 1 GB/s of throughput per year is:

(188/4 + 15*4.4) / 13.7 = 8.25

USD

For the DDR5 RAM, the cost per 1 GB/s of throughput per year is:

(25/4 + 10*4.4) / 72 = 0.70

USD

The results show that the DDR5 RAM is significantly more cost-efficient than the NVMe SSD when considering the cost per 1 GB/s of throughput per year. This finding supports the idea that, in the absence of throughput limitations, using high-throughput RAM can be more economically viable for mining than using NVMe SSDs, despite the difference in storage capacity.

Conclusion

The analysis of the shrink and Moore attacks on the PoRA consensus mechanism highlights potential vulnerabilities in the system. The article demonstrates that without proper limitations on mining rewards and a sufficiently high number of random accesses, attackers could gain economic benefits by using cheaper, smaller storage devices or leveraging advancements in technology to increase throughput. To mitigate these risks, the PoRA mechanism should be designed with appropriate parameters, such as limiting mining rewards and ensuring a high number of random accesses. Additionally, the comparison between NVMe storage and RAM suggests that RAM-based mining rigs could pose a significant threat to the security of the system, as they are more cost-effective per unit of throughput.

Further research

We are planning to publish soon an article with green (no PoW inside) proofs of storage, based on statistics, economics, and zkSNARK cryptography, suitable for our decentralized storage research, available at:

- Blockchain Sharded Storage: Web2 Costs and Web3 Security with Shamir Secret Sharing
- Minimal fully recursive zkDA rollup with sharded storage

Links

- Qi Zhou. Decentralized Storage on Large Dynamic Datasets with Applications for Large Decentralized KV Store.