TL;DR

Actual simulations were performed on the cost of attacks against the DA Layer and the probability of being attackable.

1. The cost of an attack on the DA Layer is totally unlike that of a PoS double vote.
2. The probability of attackability increases rapidly when the degree of redundancy of data holding nodes falls below a certain number like 200.

Withhold Attack:

The block withholding attack is always the biggest issue in Layer 2. If finality is confirmed in a state where the block data or part of it is withheld, in the case of finality using zkp, everyone's assets will be frozen. And in a security model like ORU that goes through a fraud proof period, the attacker can extract all the assets.

As a current countermeasure, DAS is basically used to prevent withhold attacks in the DA Layer. By having randomly selected nodes hold finely chunked data in a redundant manner and having validators verify it with KZG commitments, etc., it becomes extremely difficult to lose a block. However, while it is cryptographically possible to prove data possession, it is not possible to prove that withholding was not done.

BP/Validator Collusion Economics:

The economics of block producers and validators for blocks containing directly verified data differs completely before and after the separation of the DA Layer. For example, in the previous Ethereum or Bitcoin, the probability of blocks continuing after a block containing malicious transactions was extremely low. This is because subsequent BPs would surely confirm that the content is malicious by verifying it, and fork. In other words, to be malicious, you need to deceive future validators regarding the transactions. However, for blocks with a separated DA area that is not directly verified in the future, there is no need to deceive future validators in order to pass blocks with withheld data.

Incentives for Validator Collusion

Pros:

1. Can steal all assets of ORU
2. Can receive block rewards

Cons:

1. Reputation gets bad

In other words, unlike the case of Ethereum PoS, which uses the term "slashing" in the same way, attackers can obtain rewards while also obtaining attack gains, and the cost is not defined in the protocol. Basically, the economic security of slashing cannot be applied to the incentives for data retention and attack, and it will depend on non-incentivized honesty assumptions or community reputation.

Even so, if this attack can be established with 1/N honest security, it can be said to be sufficiently prevented. Is the security assumption of the DA Layer 1/N?

1/N Security Assumption of DAL

In DAS, since data is chunked in 2D and distributed redundantly, colluding to withhold seems to be very troublesome. However, considering that you only need to be able to withhold even a part of the block data to perform a DA attack, the assumption of DA attack in DAL including DAS can and should be simplified and calculated as follows:

Let's assume a protocol where a single chunk of data like a block is recovered from m redundant nodes, and n such data chunks are posted over a certain period of time, say a year. (Note that this m is neither the total number of nodes holding chunked data in DAS, nor the number of light clients in Celestia!)

Let's say that a ratio s (0<s<1) of m are fully aligned nodes, i.e., always online and never malicious or fooled. For DA attacks, unlike explicit double voting in PoS or DPoS, you can contribute to the attack just by going offline for a few days. Therefore, it is reasonable to count nodes that are not honest and online as attackers, rather than considering what percentage of attackers there are.

Let's say the threshold for block production is t (0<t<1), where everyone's DAS proof and signatures need to be collected. (If all are required, a liveness attack that simply stops the blockchain becomes possible. You need a threshold.)

What do the attackers need to do? The attackers succeed in the attack only if they manage to assign all mt of the nodes selected for block production from m to the m

(1-s) group. It is a kind of 1/N security, but if they succeed even once in n repetitions of this game, the entire attack succeeds.

attackrisk_1: 1 - (1- {}*{m(1-s)}\mathrm{C}*{mt} / {}*m\mathrm{C}*{mt})^n

This can be expressed with the probability distribution function of hypergeometric distribution f(k,N,K,n) as well.

attackrisk_1: 1 - (1- f(mt,m,m(1-s),mt))^n

Let's actually plug in some typical values. Let's assume we want to guarantee security for 3 years. s is conservatively about 1/5, or optimistically 1/3. Let's use 1/5 here. Assuming that 2/3 of the nodes are fully aligned (never go offline or absolutely never join an attack) for a long period of time is extremely scary. With a simple block interval of 15 seconds, n is 6307200. Let's set the threshold t to 1/2.

For example, if m is 100,

attackrisk_1: 1 - (1- {}*{80}\mathrm{C}*{50} / {}*{100}\mathrm{C}*{50})^{6307200} = 0.42

You can use this python code to calculate it or play with numbers.

import math m=100; s=0.2; t=0.5; n = 6307200 m1s = int(m*(1-s)); mt = int(m*t) attackrisk = 1 - (1 - float(math.comb(m1s, mt))/float(math.comb(m, mt)))**n print(attackrisk)

There is a 42% chance of being attacked once within 3 years.

This is not a double spend attack, but in the case of ORU, all funds are completely drained. I just put the normal numbers which you can come up with easily.

And when m=50, it is almost 100% likely to be attacked.

When m=200, it can be kept to a practically safe probability of around 4.2*10^-8, showing how crucial it is to increase the number of m even if it's not a significant difference

. With this model, it can be shown that security improvement by chunking is effective, and also that it is barely above the passing line.

In the case of chunking data like DAS, m increases but n also increases greatly, and security improvement is only possible if the increase in m cancels out the increase in n. For any blockchain system, including Ethereum, there has been no successful case of controlling the number of nodes as desired in the long term. If we expect an increase in m by reducing the load, we need to try various models to look at the relationship between data size and available nodes (= the relationship between c and c' described later).

Here is a favorably interpreted model for the case where data size is reduced by chunking:

attackrisk_2: 1 - (1- {}*{m(1-s)c'}\mathrm{C}*{mtc'} / {}*{mc'}\mathrm{C}*{mtc'})^{nc}

where c is the number of divisions by chunking, and c' is the term representing the increase in m due to that.

attackrisk_2 < attackrisk_1

when

(1- {}*{m(1-s)c'}\mathrm{C}*{mtc'} / {}*{mc'}\mathrm{C}*{mtc'})^{c} > 1- {}*{m(1-s)}\mathrm{C}*{mt} / {}*m\mathrm{C}*{mt}

While the notation itself is simple, it seems very difficult to back-calculate the proper relationship between c and c' from this inequality. (We can express {}*{m(1-s)}\mathrm{C}*{mt} / {}*m\mathrm{C}*{mt}

as the cumulative distribution function of the hypergeometric distribution and approximating it with the Poisson distribution using the fact that n is large enough. But it introduces Γ function, ruining all the appetite for wonder.)

Since it is difficult to derive a general solution, let's plug in various values based on m=200, s=0.2, t=0.5. This program plots the relationship between c and c' when attackrisk_1 and attackrisk_2 are equal under the above conditions.

c: 2 c': 1.0

c: 5 c': 1.02

c: 10 c': 1.05

c: 50 c': 1.15

c: 200 c': 1.16

c: 500 c': 1.17

c: 600 c': 1.17

c: 700 c': 1.17

The code is here (547 Bytes)

Even if one data is divided into c=700, c' is only about 1.2, meaning m increases by only about 20%, which is of order log(n). Assuming that the total number of nodes is divided by the number of groups c since the data is divided by c, the total number of nodes turns out to be t_o = c log(c) * m

. (Otherwise, there is no point in chunking if security only decreases.)

First, it seems reasonable to try a lognormal distribution for the relationship between data size d = d' / c

(d' is the origial data size) and probability of node, which relates to t_o

. This is the distribution that shows the relationship between the amount of wealth and the number of people for that wealth in the world, so it makes sense to apply this distribution if we consider storage as wealth. The probability density function of the lognormal distribution looks close to the inverse function of log(x)/x, so we can expect the number of nodes needed to maintain security after chunking almost follows this distribution. Conversely, if we assume this distribution, it does not seem security has improved significantly.

And, please note that this expectation can be said on the favorable model.

Inference and Conclusion

DA Layers are fine.

A DA Layer with a small m is spicy if it carries an ORU where DA attacks can be incentivized. We should also be careful about cases where trying to improve security ends up increasing n and becoming more dangerous. Even if fancy cryptography is used to create a DA, we still need to know what m and n are, and it solves nothing if they are bad.

Regarding chunking, assuming that the increase in nodes follows a lognormal distribution, it may be admitted that it is useful, but if m falls below a number like 200 and is small, any effort is futile.

By the way, what would be a good term to call this security assumption of "in order to attack, you have to happen to be assigned all the bad actors to all the seats" instead of calling it a 1/N honesty assumption?