



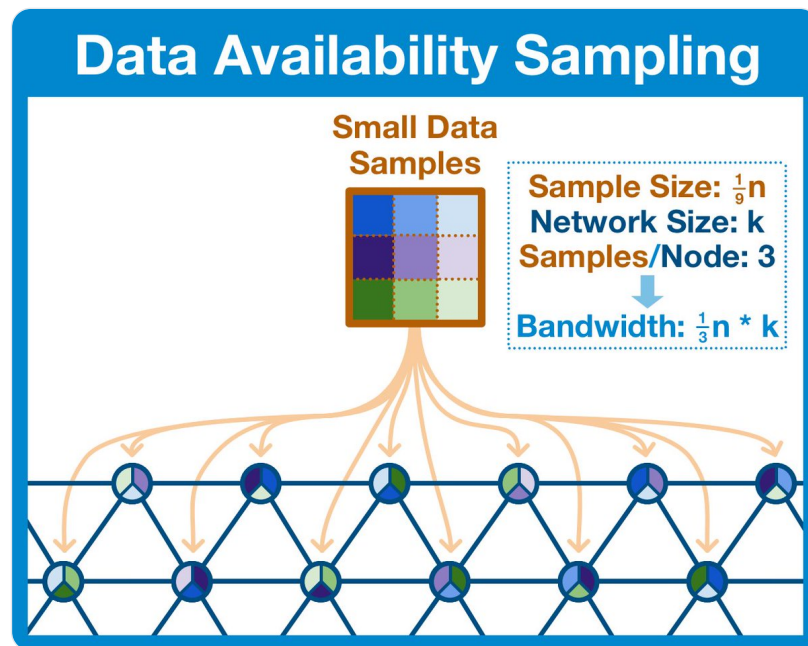
Haym @SalomonCrypto

Oct 24 · 22 tweets · [SalomonCrypto/status/1584559535959658496](#)

Tr

(1/21) The World Computer of tomorrow is a rollup-centric [@ethereum](#). But rollups produce data, lots and lots of data. How can we scale a growing network without growing bandwidth?

We've talked about randomly sampled committees, now it's time to talk Data Availability Sampling.



(2/21)

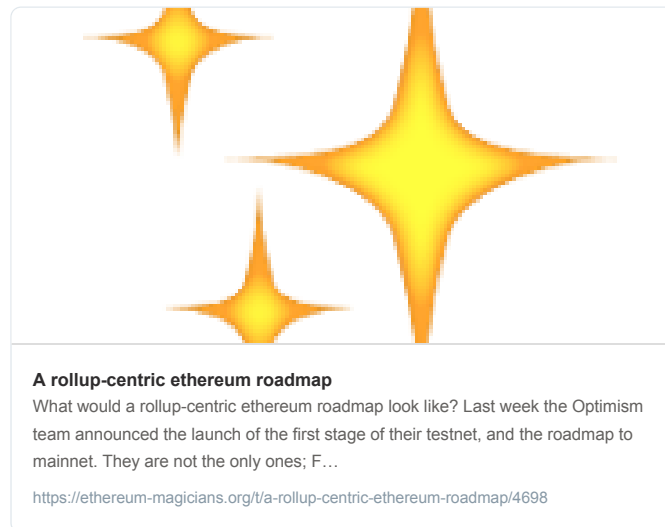
< NOTE >

If you've already read my post on Randomly Sampled Committees, the following introduction is copy/pasted. If you've seen this part, skip to tweet 10.

< /NOTE >

(3/21) [@ethereum](#) has been charting a course towards a rollup-centric future for ~2 years now.

Tl;dr Ethereum will scale by offloading execution to performance-optimized blockchains, while settlement and economic security will remain with mainnet.



(4/21) The rollup-centric paradigm is based around quickly and cheaply executing transactions in a more centralized environment, and then posting a (compressed) record back to [@ethereum](#).

In the rollup-centric future, it becomes critical to ensure that data is available.

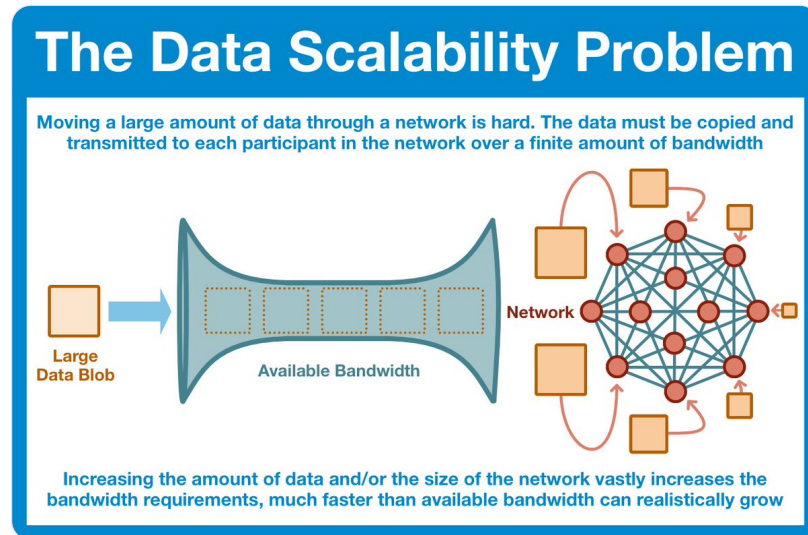


(5/21) Now look, I am an [@ethereum](#) zealot, a true believer in the inevitable dominance of the World Computer. I believe that billions and billions of transactions will happen on rollups every single day.

And that is A LOT of data circulate through the Ethereum network.

(6/21) The goal: verify the availability of high volumes of data without requiring any single [@ethereum](#) node to personally download and verify ALL of the data.

Even if we were ok with forcing every node to download the data, the reality is that the network could not handle it.



(7/21) Before we continue we need to be clear about what kind of data we are talking about.

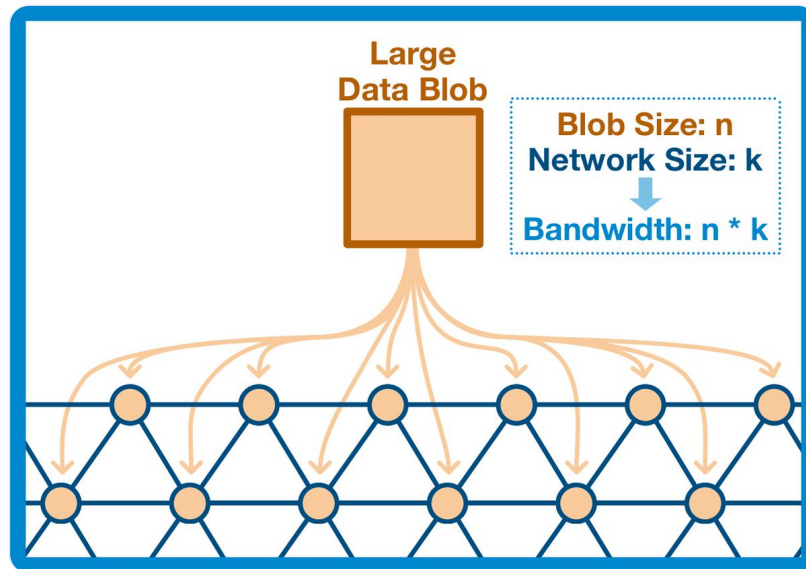
Any data within the EVM (the computing environment within [@ethereum](#)) needs to be communicated across the entire network. ALL nodes needs a copy of EVERY EVM state change.

(8/21) Rollups exists OUTSIDE of the EVM; they are independent blockchains. Yes, they post a copy of the transaction to [@ethereum](#) but that's just to ensure immutability.

We only care that a copy is posted to the World Computer, not that each node gets a copy.

(9/21) The most simple and obvious way to ensure that the data is posted to the World Computer is just to have every node in the network download it. Yes, the nodes don't NEED it, but at least we KNOW it's there.

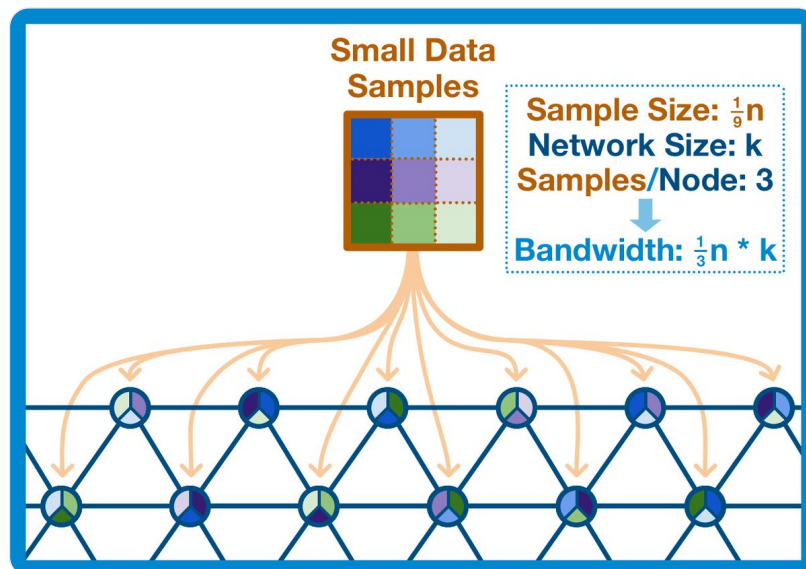
But, as we've already discussed, this just isn't scalable.



(10/21) Our goal is to guarantee that the entire blob was published to the network and is available for download... but there's no reason every node has to check the entire blob.

What if each node checked just some small random sample of the blob?

(11/21) Yes, no single node will download the entire block, but if we are careful about how we break up our blob and ensure our sampling is random enough, we can be confident the entire block is available.



(12/21) A good goal is for every node to verify that at least half of the blob's data is available. If less than half the data is available, then we can be confident that one of these samples will fail and the node will reject the blob.


But what if only ONE sample is missing?

(13/21) Let's say a blob is split into 100 samples, with each node randomly selecting 50. In this scenario, it is likely that a single invalid or unavailable sample might slip through.


We can't allow even a single txn to skip verification; what if it mints 100 trillion \$USDC?

(14/21) Fortunately, we have a simple solution: we are going to repackage our data.

Instead of publishing the raw blob, we are going to apply a technology called erasure coding. If a blob has been erasure encoded, the entire blob can be recovered with just 50% of the data.



**Haym**  
@SalomonCrypto · [Follow](#)

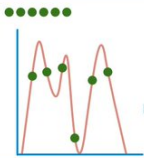


(1/19) Cryptography Basics: Polynomial Erasure Codes

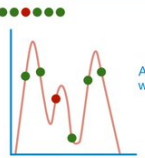
Moving data across the internet is difficult; assuming you trust all the intermediate parties, you still have vicious network conditions. How can you ensure your data stays clean?

A lesson in practical cryptography.

### Polynomial Encoding

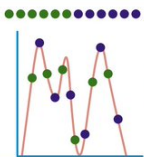


Data can be represented as a formula with a polynomial interpolation algorithm

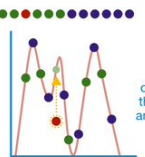


Any change in the data will result in completely different formula/polynomial


### Polynomial Error-Detection







The polynomial can generate additional points on the same curve



Changes can be detected by rebuilding the formula/polynomial and checking the points

3:59 PM · Oct 23, 2022 

[Read the full conversation on Twitter](#)

 157  Reply  Copy link

[Read 6 replies](#)

(15/21) Vitalik authored one of the better metaphors I've read for erasure coding, I'll just leave it here.

Bottom line: erasure code takes a message of length  $n$  and doubles it to  $2n$ . But this added data comes with extra functionality: the ability to recover ALL  $2n$  points.

The simplest mathematical analogy to understand erasure coding is the idea that "two points are always enough to recover a line": if I construct the "file" consisting of four points  $((1, 4), (2, 7), (3, 10), (4, 13))$ , all of which are on one line, then if you have any two of those points, you can reconstruct the line, and compute the remaining two points (we assume the  $x$  coordinates  $1, 2, 3, 4$  are a fixed parameter of the system, not the file creator's choice). With higher-degree polynomials, we can extend this idea to create 3-of-6 files, 4-of-8 files, and generally  $n$ -of- $2n$  files for any arbitrary  $n$ , with the property that if you have *any*  $n$  points of the file, you can compute the remaining ones out of  $2n$  which are missing.

(16/21) At first glance, this might not make too much sense. We are working with bandwidth issues and we just doubled the size of our blob...

But remember, we didn't just double it, we increased the size with these special new erasure codes. Let's think this through.

(17/21) First, the erasure codes change the game for an attacker; instead of hiding a single sample, an attacker would have to hide at least 50% of the blob. Otherwise, the network could just reconstruct the rest of the blob using the erasure codes.

(18/21) Now, let's think about random sampling. Each choice is going to be completely random, there is no way for anyone (malicious or otherwise) to precompute which samples are going to be requested.

So an attacker needs to hide >50% of the data, randomly selected.

(19/21) Now combine these two concepts:

- an unavailable block needs to hide >50% of the data
- no entity can know which samples are going to be requested

Random sampling becomes incredible efficient.

(20/21) The example that Dankrad always gives is that if we query 30 random samples from an [@ethereum](#) block and all are available, the probability that less than 50% are available is  $2^{-30}$  or .000000093%.


Our bandwidth problem looks WAY different at 30 samples than a full blob.

(21/21) As [@ethereum](#) continues to build towards a rollup-centric future, it becomes critical to design for increased data requirements of a robust rollup ecosystem.


Data availability sampling is a huge step forward in ensuring data is available without crushing the network.

Like what you read? Help me spread the word by retweeting the thread (linked below).

Follow me for more explainers and as much alpha as I can possibly serve.



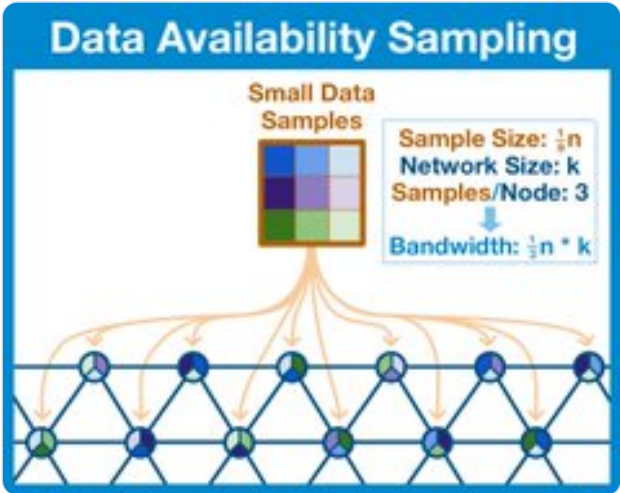
**Haym**  
@SalomonCrypto · [Follow](#)



(1/21) The World Computer of tomorrow is a rollup-centric [@ethereum](#). But rollups produce data, lots and lots of data. How can we scale a growing network without growing bandwidth?

We've talked about randomly sampled committees, now it's time to talk Data Availability Sampling.


### Data Availability Sampling




Small Data Samples

Sample Size:  $\frac{1}{3}n$   
Network Size:  $k$   
Samples/Node: 3  
↓  
Bandwidth:  $\frac{1}{3}n * k$

2:56 PM · Oct 24, 2022

[Read the full conversation on Twitter](#)



...