



# Dencun Hardfork

# Igor Mandrigin

2

- CTO & Cofounder, **Gateway.fm**
- core dev, Polygon's zkEVM & Gnosis Chain
- previously
  - core dev & researcher, Ethereum (Erigon Team)
  - fintech, Opera
  - mobile browsers, Opera



Presto / RPC / Staking / Core R&D



PALM NFT STUDIO

WIREX LUKSO



**Best in class**  
technical support

Dev experience in  
**blockchain**  
**protocols**

**99.99%**  
uptime

full **CI/CD**

**24/7/365**  
on-call



# cdk-erigon

4



**Eduardo Antuña Díez** 22:41

zkEVM Cardona cdk-erigon beta release

Happy to announce that we already have the **first beta release** of [cdk-erigon](https://github.com/OxPolygonHermez/cdk-erigon) compatible with the zkEVM and available for anyone who want to start testing it on their infrastructure: <https://github.com/OxPolygonHermez/cdk-erigon/releases/tag/v0.9.4>. The instructions on how to run it are in the readme of the repository, but feel free to ask any questions you may have.

For now it is **only recommended** to use it for **Cardona testnet**, but soon it will be available for mainnet. We're still using the zkevm-node as RPC on our infra, but soon we will start using [cdk-erigon](https://github.com/OxPolygonHermez/cdk-erigon) as official RPC endpoint.

Special thanks to all the [gateway.fm](https://gateway.fm) team for making this possible



# The Talk

5

- Intro
- DA, L2s & Blobs
- KZG & Scalability
- Q&A





# Introduction

# Dencun Hardfork

7

**Deneb** (Consensus Layer)

**Cancun** (Execution Layer)

## Included EIPs

- EIP-1153: Transient storage opcodes
- EIP-4788: Beacon block root in the EVM
- EIP-4844: Shard Blob Transactions
- EIP-5656: MCOPY - Memory copying instruction
- EIP-6780: SELFDESTRUCT only in same transaction
- EIP-7044: Perpetually Valid Signed Voluntary Exits
- EIP-7045: Increase Max Attestation Inclusion Slot
- EIP-7514: Add Max Epoch Churn Limit
- EIP-7516: BLOBBASEFEE opcode





## DA, L2s & Blobs



# Dencun Hardfork

9

 **Jesse Pollak (jesse.xyz)**   @jessepollak · Mar 14





after 2 years of hard work, blobs are now live on @base

before: \$0.31

after: \$0.00 (but actually \$0.0005)

...

[Show more](#)

 <b>Ethereum</b> <span>\$4.01</span> 0.001 ETH	 <b>Ethereum</b> <span>\$4.01</span> 0.001 ETH
↓	↓
 <b>vault.jesse.xyz</b> 0x8E86...5e49	 <b>vault.jesse.xyz</b> 0x8E86...5e49
<b>Wallet used</b> jesse.xyz 0x8491...8bf1	<b>Wallet used</b> jesse.xyz 0x8491...8bf1
<b>Network</b> Base	<b>Network</b> Base
<b>Network fee</b> ⓘ <span>\$0.31</span> ⚙️	<b>Network fee</b> ⓘ <span>\$0.00</span> ⚙️
New quote in: 17	New quote in: 23
<b>Total cost</b> \$4.32	<b>Total cost</b> \$4.01



# Dencun Hardfork

10

**Jesse Pollak (jesse.xyz)** · Mar 14  
after 2 years of hard work, blobs are now live on @base

before: \$0.31  
after: \$0.00 (but actually \$0.0005)  
...  
[Show more](#)

Item	Before	After
Ethereum	\$4.01 0.001 ETH	\$4.01 0.001 ETH
Wallet used	jesse.xyz 0x8491...8bf1	jesse.xyz 0x8491...8bf1
Network	Base	Base
Network fee	\$0.31	\$0.00
Total cost	\$4.32	\$4.01

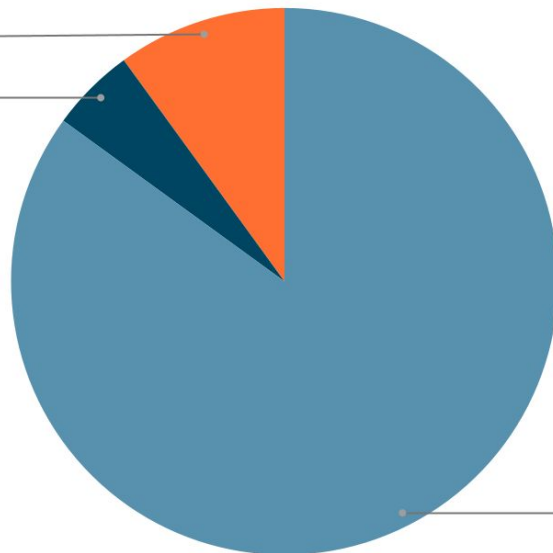
## Costs of L2

Infra

10.0%

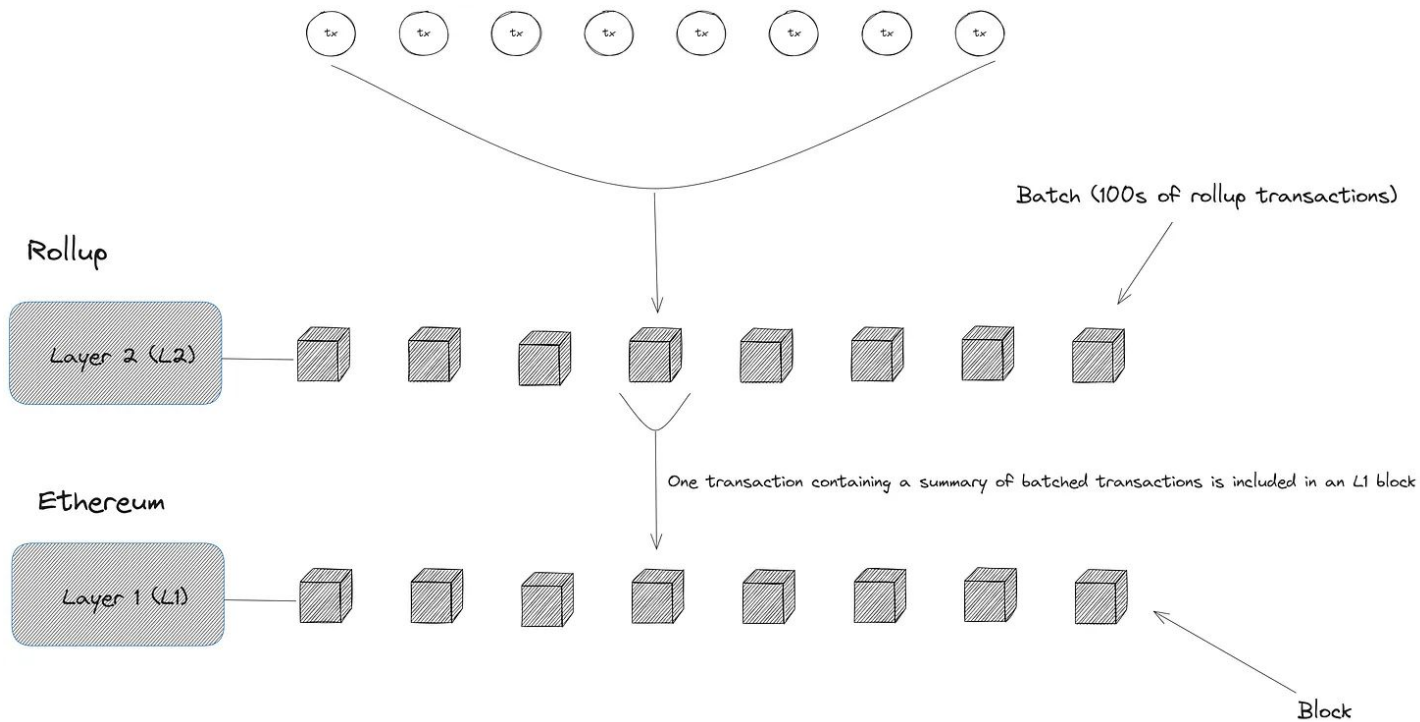
Proofs

5.0%



# Data Availability

11



# Data Availability

12

From: [0x148Ee7dAF16574cD020aFa34CC658f8F3fbd2800](#) (Polygon: zkEVM Batch Sequencer)

Interacted With (To): [0x519E42c24163192Dca44CD3fBDCEBF6be9130987](#)

ERC-20 Tokens Transferred:

All Transfers Net Transfers

From Polygon: zkEVM Batch Sequencer To Polygon: Polygon zkEVM Proxy For 0.8 (\$0.80) Polygon Ecos...

Value: 0 ETH (\$0.00)

Transaction Fee: 0.04259360535141426 ETH (\$151.69)

Gas Price: 32.781358115 Gwei (0.000000032781358115 ETH)

Gas Limit & Usage by Txn: 1,460,803 | 1,299,324 (88.95%)

Gas Fees: Base: 29.608571677 Gwei

Burnt Fees: Burnt: 0.038471127785646348 ETH (\$137.00)

Other Attributes:

Txn Type: 0 (Legacy) Nonce: 57032 Position In Block: 75

Input Data:

Function: sequenceBatches((bytes,bytes32,uint64,bytes32)[], uint64, uint64, address)

#	Name	Type	Data
0	batches.transactions	bytes	0xb000000020000000f901d41b85010fcc140083039b2094678aa4bf4e210cf2166753e054d5b7c31cc7fa8687
0	batches.forcedGlobalExitRoot	bytes32	0x00
0	batches.forcedTimestamp	uint64	0
0	batches.forcedBlockHashL1	bytes32	0x00



# Data Availability

13

0x148Ee7dAF16574cD020aFa34CC658f8F3bd2800 (Polygon: zkEVM Batch Sequencer) 

 0x519E42c24163192Dca44CD3fBDCBEF6be9130987 

All Transfers Net Transfers

From Polygon: zkEVM Batch Sequencer To Polygon: Polygon zkEVM Proxy For 0.8 (\$0.80)  Polygon Ecos... (POL)

0 ETH (\$0.00)

0.04259360535141426 ETH (\$151.69)

32.781358115 Gwei (0.000000032781358115 ETH)

1,460,803 | 1,299,324 (88.95%)

Base: 29.608571677 Gwei

Burnt: 0.038471127785646348 ETH (\$137.00)

Txn Type: 0 (Legacy) Nonce: 57032 Position In Block: 75

Function: sequenceBatches((bytes,bytes32,uint64,bytes32)[], uint64, uint64, address)

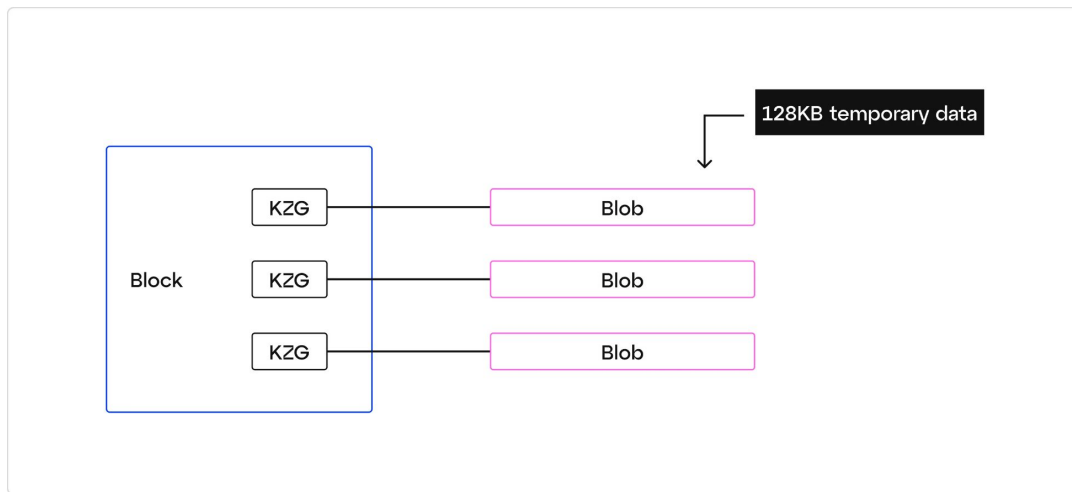
#	Name	Type	Data
0	batches.transactions	bytes	0x0b00000002000000f901d41b85010fcc140083039b2094678aa4bf4e210cf2166753e054d5b7c31cc7fa8687
0	batches.forcedGlobalExitRoot	bytes32	0x00
0	batches.forcedTimestamp	uint64	0
0	batches.forcedBlockHashL1	bytes32	0x00

~10kb calldata

RESTO  
by gateway.fm

# Blobs

14









# Blobs



## Block #32930553

Validated by  0xc6Cda6a69DDc8200Da1140546dF8E830F5e7827E 

Details Transactions **Blob txns** Withdrawals


Txn hash	Type	Method	From/To	Value XDAI	Fee XDAI
 0xefa7d0b314...7cd3 1w ago	<b>Blob txn</b>  Success	0x8d7cd6da	↓  0xCA...Fe26   PublicBlobFeed 	0	0.00004902

## Transaction details

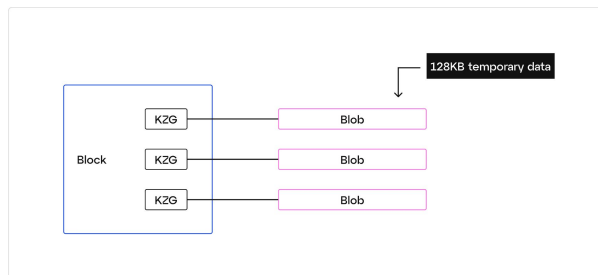
  0xCA...Fe26  called 0x8d7cd6da on  PublicBlobFeed  

 2 Explorers

Details Token transfers User operations Internal txns **Blobs** Logs State Raw trace

Blob hash	Data type	Size, bytes
0x01fdb1d171b2de0b6a72c89751586165adce12f80cf4237f5ef44bf11e79e39 	 Image	131,072

<https://gnosis.blockscout.com/tx/0xefa7d0b31405366a00902d0e5ce70d7924751772f6244519029bf091dd607cd3?tab=blobs>



	Blockspace	Blobspace
Seen by all nodes	Yes	Yes
Storage	Execution Client	Consensus Client
EVM access	Yes	No
Longevity	Forever	18 days
Cost	Expensive	Cheap





```
const response = await fetch(network.beacon + `/eth/v1/beacon/blob_sidecars/${slot}`);
const data = await response.json();

if (address !== currentAddress) {
    return;
}

for (const blob of data.data) {
    const hash = await toVersionedHash(blob.kzg_commitment);
    if (hash === versionedHash) {
        handleBlob(blob.blob, address, metadata);
        return;
    }
}
```



# Blobs Send (Execution Layer)

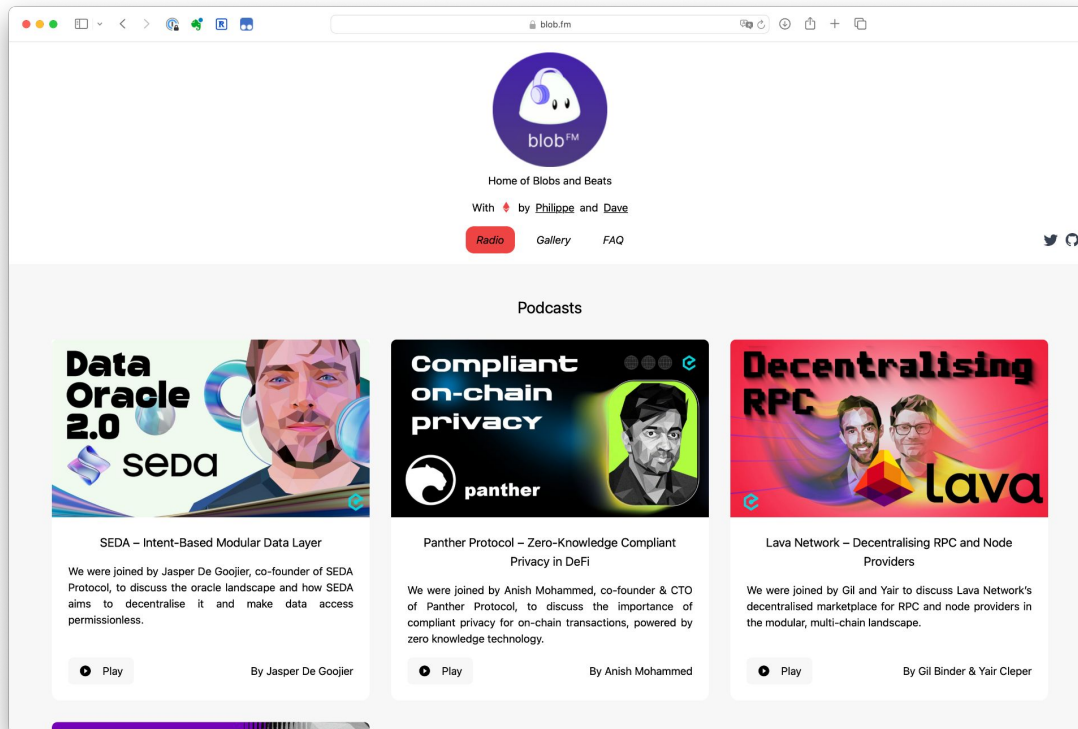
18

```
// Blob
const blobs = [createBlob(blob)];
const kzgCommitments = blobsToCommitments(blobs);

// Transaction
const txData: BlobEIP4844TxData = {
  data: '0x8d7cd6da',
  gasLimit: 50000,
  to: Address.fromString(network.gallery),
  chainId: network.id,
  maxFeePerBlobGas: 50n * oneGwei,
  maxPriorityFeePerGas,
  maxFeePerGas: maxPriorityFeePerGas + (feeData.maxFeePerGas ?? 49n * oneGwei) +
    blobs,
  kzgCommitments,
  blobVersionedHashes: commitmentsToVersionedHashes(kzgCommitments),
  kzgProofs: blobsToProofs(blobs, kzgCommitments),
  nonce
};

const blobTx = BlobEIP4844Transaction.fromTxData(txData, { common });
const signed = blobTx.sign(hexToBytes(wallet.privateKey));
const raw = bytesToHex(signed.serializeNetworkWrapper());
tx = (await provider.send('eth_sendRawTransaction', [raw])) as string;
```







# KZG, Scalability & Danksharding

 CHANGED 2 YEARS AGO



 Like

 Bookmark

 Subscribe

## New sharding design with tight beacon and shard block integration

Previous data shard construction: Add  $n = 64$  data shards to the beacon chain. Each slot,  $n$  proposers independently propose their shard blobs, which are then confirmed by committees. Once a shard blob is confirmed (this can take several slots), it can be referenced by the execution chain.

Here is an alternative proposal: Add a new type of transaction that can contain additional sharded data as calldata. The block is then constructed by a single block builder (which can be different from the proposer using proposer builder separation), and includes normal transactions as well as transactions with sharded calldata. This is efficient because it means that tight integrations between rollups and L1 become possible, and it is expected that this “super-block-builder” strategy will emerge in practice anyway in order to maximize MEV extraction.



The work that is already done in this EIP includes:

- A new transaction type, of the exact same format that will need to exist in “full sharding”
- *All* of the execution-layer logic required for full sharding
- *All* of the execution / consensus cross-verification logic required for full sharding
- Layer separation between **BeaconBlock** verification and data availability sampling blobs
- Most of the **BeaconBlock** logic required for full sharding
- A self-adjusting independent base fee for blobs

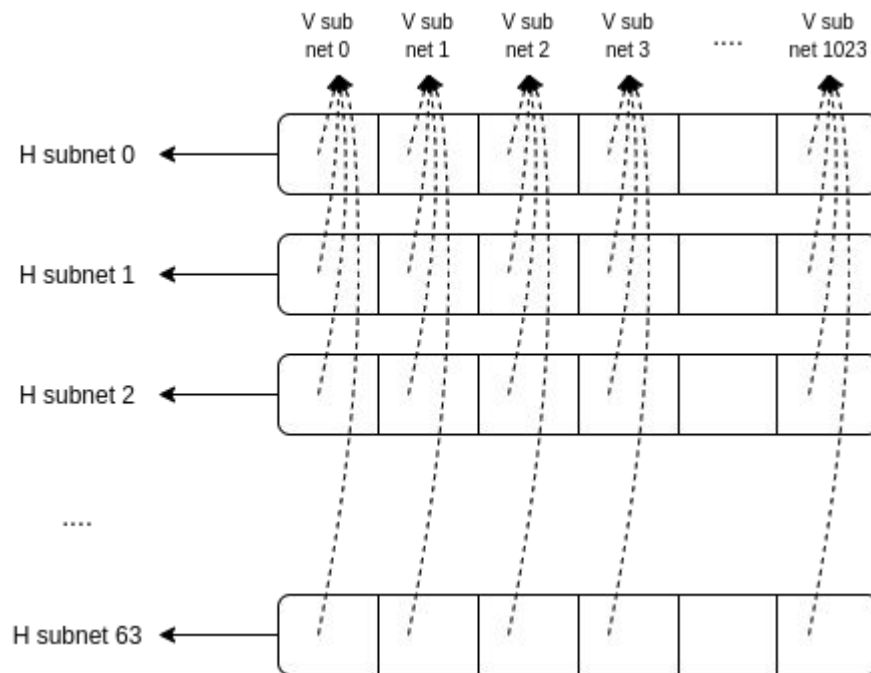
The work that remains to be done to get to full sharding includes:

- A low-degree extension of the **commitments** in the consensus layer to allow 2D sampling
- An actual implementation of data availability sampling
- PBS (proposer/builder separation), to avoid requiring individual validators to process 32 MB of data in one slot
- Proof of custody or similar in-protocol requirement for each validator to verify a particular part of the sharded data in each block



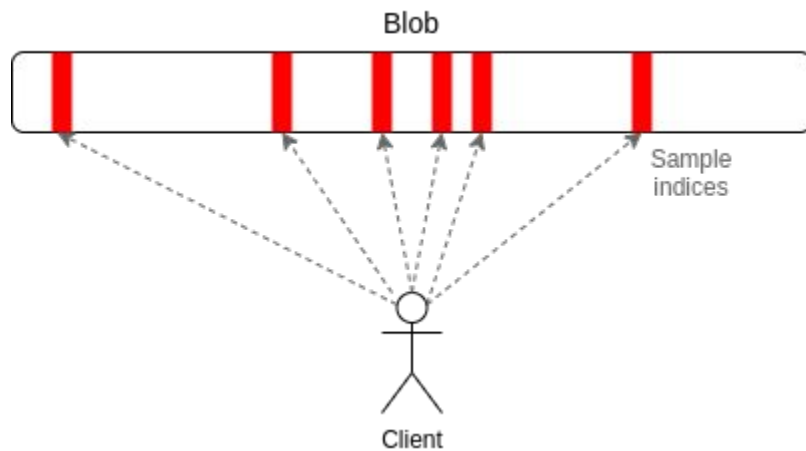
# Danksharding

23



# Danksharding

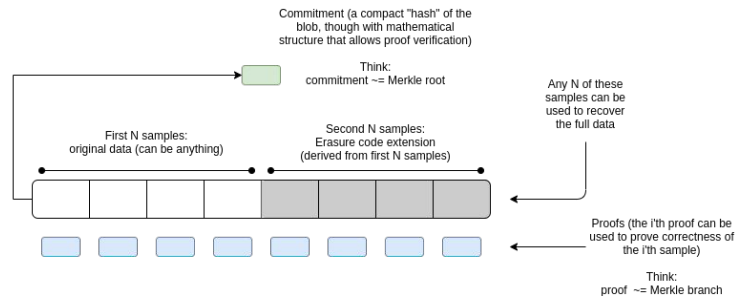
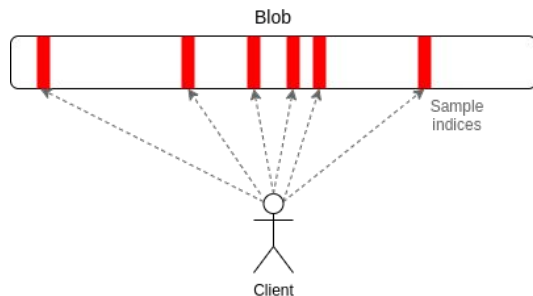
24





# KZG (non-cryptographer's view)

25



## KZG Commitment Scheme

First, the prover commits to data by creating a point on the elliptic curve. If the data changes, the prover cannot create valid proofs.

**Prover**



1)

Commit

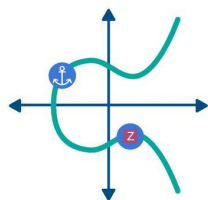


3)

Proof



Evaluation



**Verifier**



2)

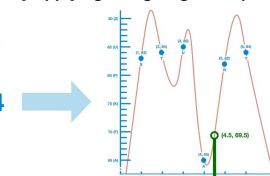
Request

Next, the verifier gives a data point. The prover builds a new elliptic curve point and a polynomial evaluation around that point.

## Polynomial Commitments

Encode data to polynomial form by applying a Lagrange Interpolation

S T U A R T  
83, 84, 85, 65, 82, 84  
1 2 3 4 5 6



Derive a new, non-sensitive point on the polynomial

PRIVATE  
PUBLIC

Post the commitment to this specific polynomial (and therefore to the original data)

[4.5, 69.5]





Q&A



# Thank you!

ask any questions here