# Blockchain Development and Fintech Final Project

Malik DAHMANI, Noah PETER, François YU

June 11, 2024

# Contents

# 1 Introduction

One of the primary challenges facing blockchain technology today is the need to reduce gas fees and increase transaction speed. EIP-4844, also known as proto-danksharding, represents a recent and significant advancement aimed at addressing these issues. This proposal introduces blob-carrying transactions, which are designed to efficiently handle large amounts of data, thereby reducing gas costs and enhancing the overall transaction throughput of the Ethereum network.

This report provides a comprehensive overview of the current state-of-the-art developments related to EIP-4844. It focuses on the theoretical aspects and potential impacts of proto-danksharding on blockchain scalability and efficiency. No experimental data or empirical results will be presented; instead, the report will offer an in-depth analysis and explanation of the concepts, mechanisms, and expected benefits of this new proposal. The goal is to elucidate how EIP-4844 could transform the Ethereum network by making transactions cheaper and faster, paving the way for more scalable and cost-effective blockchain solutions.

# 2 What is a rollup in blockchain ?

A rollup is a L2 blockchain that processes transactions separately from the main L1 blockchain in order to make each transaction faster and cheaper. It is recognized and widely used : used in Ethereum. The main idea are:

- The user bridge onto the L2 network
- Make the intended transaction on this network using the original L1 network currency.
- Transactions are processed on the rollup chain.
- A single party called "sequencer" confirms transactions, constructs L2 blocks and send the transaction and proofs to the L1 chain.
- In order to reduce gas fees, multiple transactions are submitted in batches to the L1 network.
- If needed the user can bridge back his fund to the main L1 network.

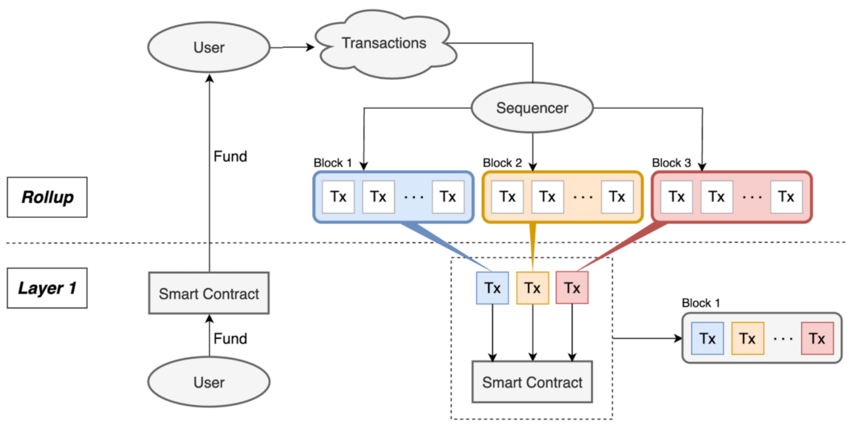



Figure 1: Interaction between rollup and Layer 1

There are two main types of rollups : Optimistic and Zero-knowledge.

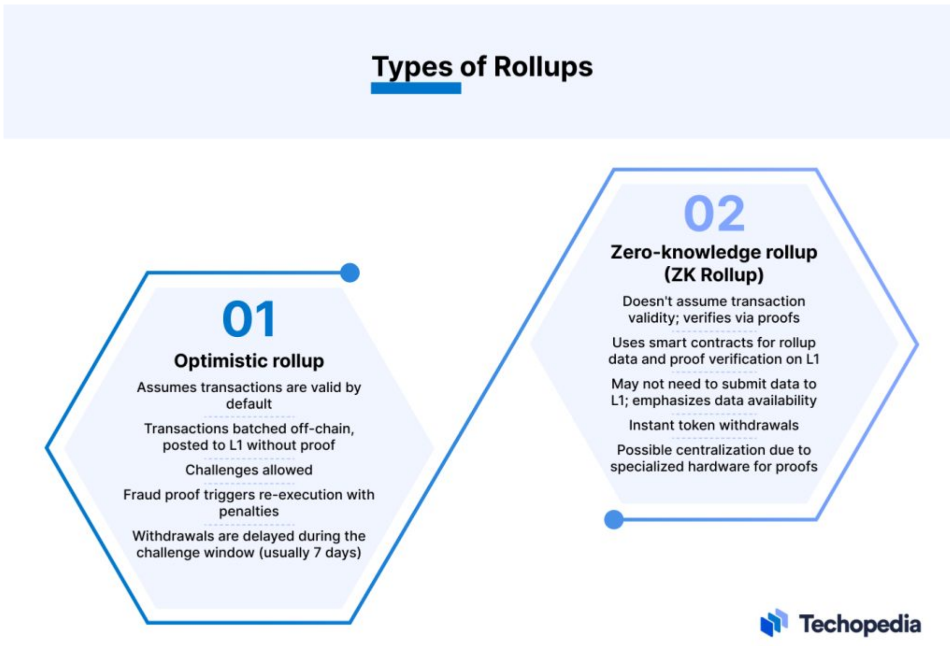



Figure 2: The different types of Rollups

## 2.1 Optimistic rollup

Optimistic rollup assumes that all the transactions that go through the chain are valid. Therefore, it is very quick as we assume that every transaction is correct and there is no need to check each of them. To fight against fraudulent transactions, each user are given a certain amount of time to contest them. The drawback is that it takes more time to withdraw funds due to the challenge window.

## 2.2 Zero-knowledge rollup

Zero-knowledge rollup verifies the validity of each transaction using cryptography method (zero knowledge proof) to do so. It uses a minimal amount of information and makes it fast and cheap. Users can quickly withdraw their funds without the 1 week delay. The data of the transaction are kept on the L2 chain while the proof is sent to the L1 layer. The disadvantage is that it may lead to centralization as producing validity proofs requires specialized hardware.

### 2.3 Advantage and Disadvantage of rollups

Optimistic rollup are faster but there is a delay to get back the funds. It means that if you want to have a quick transaction without the need of cash out quickly

Zero-knowledge rollup are more secure but more complex and can lead to centralization. Thus you need security or quick cashout.

## 3 EIP-4844

### 3.1 What is EIP-4844?

It introduces blob-carrying transaction which is a specialized transaction that includes a large data structure called a "blob." This blob is not accessible by the Ethereum Virtual Machine (EVM) during transaction execution, but its commitment (a cryptographic hash) can be accessed. This allows the network to handle large data payloads more efficiently. The transaction information are roughly the same than the previous version, it contains additional information about blob management.

### 3.2 Why use blob?

Without usage of blob, in current rollup mechanisms (Optimistic and ZK rollups), transaction data is stored in calldata, which is costly and inefficient. Although reducing calldata prices may seem like a way to improve scalability, it would significantly increase the maximum block size, causing network congestion. For instance, cutting calldata costs by tenfold could lead to worst-case block sizes of 18 MB, which Ethereum cannot support. The current gas pricing scheme ties average and worst-case loads together, requiring gas prices to be based on worst-case scenarios. This causes an average load that is less than what the network can handle.

Additionally, the average block size presents its own problems. EIP-1559 sets the target gas amount per block at 15 million gas, suggesting that the average block size that doesn't negatively impact the network should be around 1 MB (15M gas / 16 gas per byte). However, the current average block size is only about 90 kB, which is far below this optimal level. This discrepancy highlights why rollups need blob transactions.

Blobs consist of 4096 fields, each 32 bytes in size. By limiting each block to a maximum of 16 blobs, the worst-case block size remains manageable at about 2.1 MB. This setup allows the average block size to increase to around 1 MB, aligning better with the network's capacity. Moreover, unlike calldata, which is used for transaction execution on the Ethereum Virtual Machine (EVM), blobs are designed for simple storage. This design makes blobs significantly cheaper than an equivalent amount of calldata, providing a cost-effective solution to data handling.

Proto-danksharding and EIP-4488 propose creating a multidimensional fee market to separate average and worst-case loads, allowing higher average data loads without compromising network stability. By implementing these changes, the Ethereum network can avoid the current load mismatch, increasing both scalability and cost-efficiency.

| | **Average case block size** | **Worst case block size** |
|---|---|---|
| **Status quo** | 85 kB | 1.8 MB |
| **EIP-4488** | Unknown; 350 kB if 5x growth in calldata use | 1.4 MB |
| **Proto-danksharding** | 1 MB (tunable if desired) | 2 MB |

Figure 3: Block size comparison

Blobs are designed to be temporary and are deleted after 18 days. This timeframe is deemed sufficient for rollups to verify each transaction contained within a blob. The rationale behind this 18-day lifespan is that it provides adequate time for all necessary verifications and fraud proofs to be conducted, ensuring the integrity and security of the transactions.

By deleting the blobs after this period, significant storage space on the blockchain is freed up. This reduction in storage demand helps lower gas fees, as the cost associated with storing data on the blockchain is minimized. Additionally, it improves the network's speed and efficiency, as less data needs to be processed and maintained by nodes. This approach balances the need for data availability for verification purposes with the practical requirement of maintaining a scalable and efficient blockchain infrastructure.

## 3.3 How does blob improve gas managements

### Gas fee before EIP-4844

Before EIP-4844, rollups mainly used calldata. Calldata is a specific data location in Ethereum used to pass data to smart contracts. Smart contracts can access and use this data but cannot modify it as it is read-only. Calldata is primarily used for function arguments and it was the most efficient way to pass data before EIP-4844.

When a transaction is sent to a smart contract, it contains several components, one of which is calldata. Calldata includes:

- Function Selector: The first 4 bytes identify which function of the contract is being called.
- Encoded Arguments: The remaining bytes contain the encoded arguments to be passed to the function.

For example, if you call a function transfer(address _to, uint256 _amount), the calldata might look like this:
0xa9059cbb0000000000000000000000005b38da6a701c568545dcfcb03fcb875f56beddc40
0000000000000000000000000000000000000000000000000000000000000064

- 0xa9059cbb: Function selector for transfer(address,uint256).
- 0000000000000000000000005b38da6a701c568545dcfcb03fcb875f56beddc4: Address _to
- 0000000000000000000000000000000000000000000000000000000000000064: Amount _amount (100 in decimal).

So we've seen how call data works but how is it used in rollups? Rollups use calldata to post transaction data to Ethereum main net. For optimistic rollups, calldata is used to submit the transaction data and state roots, assuming the transactions are valid but allowing for challenges. For zkRollups, call data is used to post proofs and compressed transaction data.

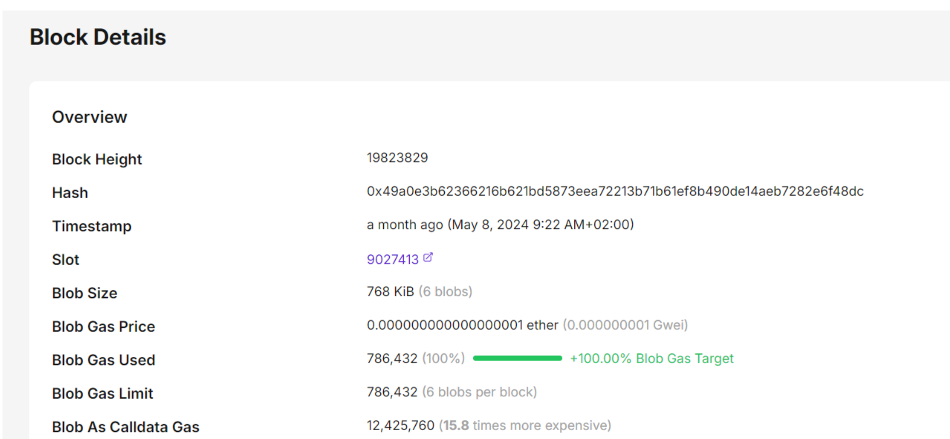
As we all know, every transaction on chain has a cost, gas cost. How is it calculated for calldata? In calldata, gas costs are calculated based on the amount of data included in a transaction. Each calldata byte costs gas, the larger the size of the transaction data, the higher the gas fees. Calldata costs 4 gas per byte equal to 0, and 16 gas for the others. For example, if a transaction includes 128 bytes of calldata, the gas cost for the calldata alone would be: $Gas\_calldata=128\times16=2048$ gas

### Gas fee after EIP-4844

EIP-4844 introduces blob-carrying transactions to address the inefficiencies associated with using calldata for large data postings. As said before, key improvements include:

- Reduced Gas Costs
- Temporary Storage
- Increased Data Availability

These transactions with EIP-4844 incorporate a new data structure called a blob. Rather than storing blobs directly within transactions, they are temporarily maintained on the consensus layer for a duration of 18 days before deletion. The design choice to exclude blobs from permanent storage on the execution layer significantly enhances gas efficiency.


**Block Details**

**Overview**

| | |
|---|---|
| Block Height | 19823829 |
| Hash | 0x49a0e3b62366216b621bd5873eea72213b71b61ef8b490de14aeb7282e6f48dc |
| Timestamp | a month ago (May 8, 2024 9:22 AM+02:00) |
| Slot | 9027413 |
| Blob Size | 768 KiB (6 blobs) |
| Blob Gas Price | 0.000000000000000001 ether (0.000000001 Gwei) |
| Blob Gas Used | 786,432 (100%) +100.00% Blob Gas Target |
| Blob Gas Limit | 786,432 (6 blobs per block) |
| Blob As Calldata Gas | 12,425,760 (15.8 times more expensive) |

Figure 4: Blob details from blobscan

To calculate, the gas reduction percentage, we used blobscan to find some informations about a blob transaction.

In the figure:

- Blob Size indicates the total size of the blobs in the block.
- Blob Gas Price indicates the price of Blob Gas.
- Blob Gas Used indicates the percentage of blobs used compared to the Target value (3 blobs)
- Blob Gas Limit indicates the maximum number of blobs (6 blobs) in a block
- Blob As Calldata Gas indicates how much gas is consumed by using calldata for the same data.

This example transaction consists of 6 blobs, about 768 KB. Uploading that data using calldata would have consumed 15.8 times more than using a blob. To calculate the gas saving:

$Gas_{savings} = TotalGas_{calldata} - TotalGas_{blob}$
$Gas_{savings} = 12425760 - 786432$
$Gas_{savings} = 11639328$

$PercentageReduction = \left(\frac{Gas_{savings}}{TotalGas_{calldata}}\right) \times 100$
$PercentageReduction = \left(\frac{11639328}{124257}\right) \times 100$
$PercentageReduction = 94\%$

This gas reduction isn't negligible and is often generalized as 90% on some popular L2 networks.

### 3.4 Impact of EIP-4844 on rollups

Some benefits of EIP-4844 were the impact on Rollups and Scalability.By bypassing Layer 1 gas competition, blob transactions enhance Ethereum's scalability. This results in more cost-effective roll-up pricing and reduced overall transaction costs as said before. Moreover, EIP-4844 introduces 1MB block sizes and blob transactions, boosting scalability and reducing rollup fees.

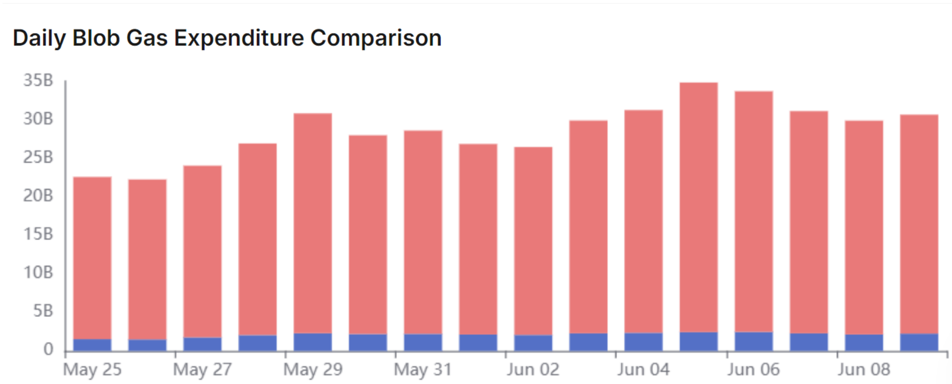



Figure 5: Daily blob gas expenditure comparison. Blue = blob gas used and Red = equivalent blob gas as calldata

To continue, the introduction of blobs allows rollups to post more data on-chain without incurring high costs. This increased capacity is critical for rollups that require high throughput and efficient data availability. In "Impact of EIP-4844 on Ethereum: Consensus Security, Ethereum Usage, Rollup Transaction Dynamics, and Blob Gas Fee Markets", they performed a detailed examination of changes in Ethereum usage by the top 10 rollups following the implementation of EIP-4844.

| | Total Data Size (MiB) | | | Total Fees Paid (ETH) | | | Price For 1MiB (ETH) | | | Total Gas Used (Gas units) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rollup Type | Before | After | Change | Before | After | Change | Before | After | Change | Before | After | Change |
| All Rollups | 0.084 | 0.183 | +116.83% | 0.075 | 0.021 | -71.38% | 1.304 | 0.231 | -82.32% | 1.725M | 0.784M | -54.53% |
| Optimistic Rollups | 0.049 | 0. 111 | +127.4% | 0.047 | 0.007 | -84.89% | 0.905 | 0.239 | -73.55% | 0.878M | 0.169M | -80.74% |
| ZK Rollups | 0.035 | 0.072 | +102.22% | 0.028 | 0.014 | -48.75% | 1.516 | 0.280 | -81.53% | 0.611M | 0.461M | -24.55% |

Figure 6: Changes of Ethereum usage by rollups before and after EIP-4844

### Gas fees for rollups

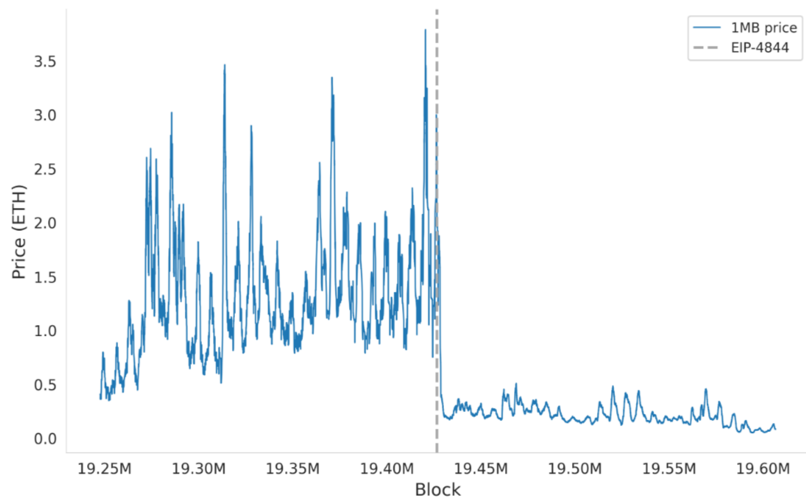



Figure 7: 1MiB price of Ethereum before and after EIP-4844

As said before, the EIP-4844 update drastically reduces the cost of gas. In the table, the total fees paid before and after the update decreased by 71.38% and the price of posting MiB reduced by 82%.

### Data size per block

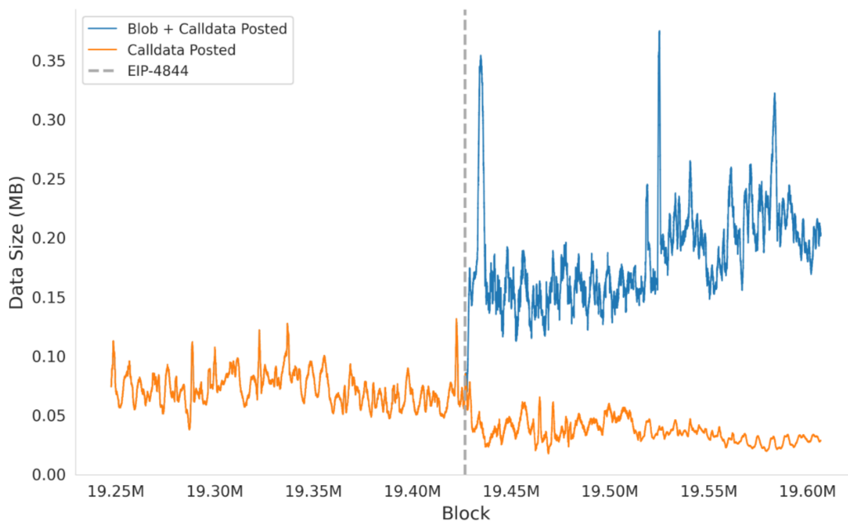



Figure 8: Data size of rollups before and after EIP-4844

The figure shows a significant increase in the data posted by rollups after EIP-4844. The data size posted increased while calldata size decreased:

- $Size_{pre-EIP-4844} = 0.084$ (average data size per block).
- $Size_{post-EIP-4844} = 0.183$ (average data size per block).

The increase in data size can be calculated as:

$$DataSizeIncreasePercentage = (\frac{Size_{post-EIP-4844} - Size_{pre-EIP-4844}}{Size_{pre-EIP-4844}}) * 100$$
$$DataSizeIncreasePercentage = (\frac{0.183 - 0.084}{0.084}) * 100$$
$$DataSizeIncreasePercentage = 117\%$$

Moreover, Calldata size decreased by 56.8%, indicating a shift towards using blobs.

## 3.5 The impact of EIP-4844 for users of rollups

| | # of transactions | | | User Delay (s) | | |
|---|---|---|---|---|---|---|
| Rollup | Before | After | Change | Before | After | Change |
| Arbitrum | 111.2 | 194.7 | **+75%** | 519.9 (197.5) | 197.5 (100.8) | **-62%** |
| Optimism | 53.8 | 91.8 | **+71%** | 55.6 (23) | 224.6 (112.9) | **+304%** |
| Base | 56.8 | 183.7 | **+224%** | 59.4 (25.6) | 161.1 (35.3) | **+171%** |
| Starknet | 18 | 31 | **+72.5%** | 17,156 (4467) | 27,675 (9504) | **+61.3%** |
| zkSync | 149.4 | 174.9 | **+17.1%** | 261.4 (115.5) | 154.9 (152.3) | **-40.8%** |
| Linea | 127.5 | 187.5 | **+46.7%** | 21,354 (24489) | 30,404 (25901) | **+42.4%** |

Figure 9: Comparison of rollup transaction numbers and user delay before and after EIP-4844

### Amount of user

EIP-4844 facilitates a higher transaction throughput for rollups by lowering the costs associated with data posting. This efficiency encourages more transactions to be processed and posted on Ethereum.

To retrieve the number of transactions posted on Ethereum, they decoded the batch transactions on Ethereum sent by rollups. This table indicates that rollups like Base experienced a 224% increase in transaction volume and that other rollups (e.g., Arbitrum, Optimism, Starknet) saw increases exceeding 70%.

### Strategy for rollup

To upload data from Rollup to L1, you must purchase the full blob making it less economical for rollups that frequently submit small batches to L1. This means that even if the uploaded data is less than 1 blob, you must use the full 1 blob. In such cases, calldata might still be more viable economically. Two strategies for rollup have been developped:

- Wait until the data can fill a blob and then buy the blob.
- Choose a frequency to release the blob quickly.

For Rollup users, they want to confirm their transactions as soon as possible while maintaining acceptable costs. For Rollup, this creates a cost and latency tradeoff: Is it better to wait until the data has accumulated to the point where a blob can be filled before purchasing it (low cost), or is it better to purchase blobs frequently and release them quickly (low latency)?

The optimal strategy for rollups would be:

- If the blob is expensive, wait longer then post the full blob.
- If the blob is cheap, quickly buy and post mostly empty blob.

### Impact on delay for user

User delay refers to the time experienced by users from the moment they initiate a transaction until it is confirmed and included in a blockchain block. This delay can be influenced by several factors, including network congestion, gas prices, and the efficiency of data management protocols.

In the figure 9, we see that most of the rollups have higher delay after EIP-4844. As said before, it can refers to the strategy from the rollup to wait that the blob is filled to be posted on chain. Thus leads for the user to maintain trust in the rollup until the transaction is committed.

# 4 Conclusion

EIP-4844 significantly improves the efficiency and scalability of rollups on Ethereum by introducing blob transactions, reducing gas costs, enhancing data management, and increasing transaction throughput. These advancements not only lower operational costs for rollups but also set the stage for future scalability solutions like full Danksharding.

However, as blob prices may increase in the future, optimizing Rollups for cost efficiency remains critical. Additionally, strategies such as blob-space sharing or selectively using calldata instead of blobs during periods of high blob prices or for small batches could be viable options under certain circumstances and specific requirements.

## 5 References

**Rollup**, Mensholong Lepcha

**What Are Rollups? ZK Rollups and Optimistic Rollups Explained**, Robert Stevens

**Quelle est la différence entre les Optimistic Rollups et les ZK-Rollups?**, Coinbase

**Ethernow website**

**Blobscan website**

**EIP-4844: Shard Blob Transactions**, Vitalik Buterin, Dankrad Feist, Diederik Loerakker, George Kadianakis, Matt Garnett, Mofi Taiwo, Ansgar Dietrichs

[**EIP-4844]2: Proto-Danksharding : A Deep Dive!**,Seongwan Park

**EIP-4844 Fee Market Analysis**, Davide Crapis

**A Comprehensive Exploration of EIP-4844 and Its Integration**, Morph

**Saving the Blobs: How The Graph Indexes and Makes Blob Data Easily Accessible**, The Graph Foundation

**EIP-4844.com**

**EIP-4844 github**, Tim BEIKO

**Impact of EIP-4844 on Ethereum: Consensus Security, Ethereum Usage, Rollup Transaction Dynamics, and Blob Gas Fee Markets**, Seongwan Park, Bosul Mun, Seungyun Lee, Woojin Jeong, Jaewook Lee, Hyeonsang Eom, Huisu Jang

**Blockchain Price vs. Quantity Controls**, Abdoulaye Ndiaye

**Analyzing and Benchmarking ZK-Rollups**, Stefanos Chaliasos, Itamar Reif, Adrià Torralba-Agell, Jens Ernstberger, Assimakis Kattis, and Benjamin Livshits

**Optimal Dynamic Fees for Blockchain Resources**, Davide Crapis, Ciamac C. Moallemi, Shouqiao Wang