

What Is EIP-4844 (Proto-Danksharding)?

The Ethereum upgrade that gave rollups their own data lane — a close read of the specification that shipped, not the roadmap it points toward.

Alain AI Lab Research · Published July 3, 2026 · 9 min read

AT A GLANCE

EIP NUMBER

4844

SHIPPED IN

Dencun · Mar 2024

BLOB SIZE

~128 KB each

TRANSACTION TYPE

0x03

LAUNCH BLOBS / BLOCK

3 target / 6 max

L2 FEE IMPACT

~10× cheaper

01

What EIP-4844 Actually Shipped

EIP-4844, formally titled “Shard Blob Transactions,” is the Ethereum Improvement Proposal that introduced **blobs** — large, temporary packets of data that ride alongside a transaction but never enter the part of the network that runs smart contracts. It is nicknamed **proto-danksharding** because it delivers the transaction format and fee machinery of Ethereum’s long-term scaling plan while deliberately leaving out the hardest cryptographic piece for later.

The distinction matters, and it is the reason this report exists as its own document.

Danksharding is the destination — a future state where the network can carry enormous amounts of rollup data cheaply. EIP-4844 is the first, concrete, already-live installment of that vision: a real specification, authored by Vitalik Buterin, Dankrad Feist, Diederik Loerakker, George Kadianakis, Matt Garnett, Mofi Taiwo, and Ansgar Dietrichs, that went into production in March 2024. Where our companion piece, [what danksharding is and where the roadmap leads](#), paints the big picture, this report reads the shipped spec line by line: the transaction type, the cryptography, the precompile, and the fee market that together made layer-2 transactions roughly ten times cheaper overnight.

The Blob-Carrying Transaction

At its core, EIP-4844 adds a new kind of Ethereum transaction. Ethereum already supports several transaction formats under a numbering scheme, and this proposal registers a fresh one: **type 0x03**, the blob transaction. A type-0x03 transaction looks much like an ordinary transfer, but it can attach one or more blobs as separate **sidecars**, data structures that travel with the transaction across the network but are stored apart from it.

Each blob is a fixed size: 4,096 **field elements**, cryptographic numbers of 32 bytes each, which works out to 131,072 bytes, or roughly 128 KB. That fixed shape is not arbitrary — it is chosen so the same data can later be sampled and verified piece by piece under full danksharding without changing the format. The single most important property of a blob is what it is *not*: blob data is not accessible to the Ethereum Virtual Machine, the environment where smart contracts execute. A contract can never read the bytes inside a blob. This is the deliberate trade that makes blobs cheap.

Blobs are also temporary. Consensus-layer nodes — the machines that secure Ethereum’s proof-of-stake network — are required to keep blob data for a minimum window of about 4,096 epochs, roughly eighteen days, after which they may prune it. That is long enough for anyone to challenge or reconstruct a rollup’s recent history, but short enough that nodes are never asked to store this firehose of data forever. Permanence was the old, expensive model; EIP-4844 replaced it with a short, guaranteed availability window.

KZG Commitments and Versioned Hashes

If contracts cannot read blob data, how does the chain prove anything about it? The answer is a **KZG commitment** — a compact cryptographic fingerprint, named for its inventors Kate, Zaverucha, and Goldberg, that stands in for the blob’s full contents. A KZG commitment is only 48 bytes, yet it binds tightly to the entire 128 KB of data: you cannot later swap the data for something else without the commitment changing.

EIP-4844 chose KZG over a simpler hash for one forward-looking reason. KZG commitments have algebraic structure that lets a validator check a small, random sample of the data and gain confidence the whole thing is available — the sampling technique at the heart of full danksharding. A plain hash offers no such shortcut. Proto-danksharding therefore lays down cryptographic scaffolding it does not yet fully use, so the network does not have to migrate later.

Because the EVM works naturally in 32-byte words and a KZG commitment is 48 bytes, the spec exposes a shorter stand-in called a **versioned hash**: a single version byte (0x01) followed by the last 31 bytes of the SHA-256 hash of the commitment, for 32 bytes total. This is the value a smart contract actually sees, retrieved through a new instruction, the **BLOBHASH opcode** (0x49), which returns the versioned hash of a blob attached to the current transaction. The version byte is a quiet piece of future-proofing — it lets Ethereum swap KZG for a different scheme one day while keeping the same 32-byte shape.

04

The Point-Evaluation Precompile

The versioned hash proves a blob *exists*, but rollups need more: they need to prove that a specific slice of that blob equals a specific value, without dragging the whole 128 KB through the EVM. EIP-4844 supplies exactly this tool as a **precompile** — a built-in contract at a fixed address, here 0x0A, that performs one cryptographic operation cheaply and reliably.

The point-evaluation precompile takes a versioned hash, a point, a claimed value at that point, the underlying commitment, and a KZG proof, and confirms they are all consistent — for a flat 50,000 gas. What it establishes is precise: a **proof of equivalence**, showing that the blob the network committed to and the data a rollup claims to have posted are one and the same. It does not, by itself, validate a rollup’s state transitions; that logic lives in the rollup’s own proving system. Both major rollup families lean on this check — zero-knowledge rollups fold it into their validity proofs, while optimistic rollups can invoke it inside fraud proofs. Readers weighing those two designs will find our breakdown of [optimistic versus zero-knowledge rollups](#) a useful companion here.

05

A Separate Fee Market for Data

The economic heart of EIP-4844 is that blobs are priced in their own currency of sorts. The spec creates **blob gas** — a distinct accounting unit, entirely separate from the normal execution gas that powers ordinary transactions. Blob gas has its own base fee that rises and falls automatically, using the same responsive mechanism as Ethereum’s main fee market: when recent blocks carry more blobs than the target, the blob base fee climbs; when they carry fewer, it falls, down to a floor of a single wei, the smallest unit of ether.

This separation is the whole point. Before EIP-4844, rollups and ordinary users bid against each other for the same block space, so a busy day in decentralized finance would directly

inflate the cost of every rollup posting its data. With two independent markets, that link is cut. For much of 2024 the blob base fee sat pinned at its one-wei floor even while regular [Ethereum gas fees](#) spiked, because blob demand stayed comfortably under target. The asymmetry runs both ways: a surge in blob usage no longer drags up execution gas, and a surge in execution gas no longer taxes rollups. Two lanes, two prices, floating independently.

The counterintuitive core: EIP-4844 made data cheaper not by compressing it, but by refusing to keep it. The old model stored rollup data in permanent, EVM-readable calldata forever. Blobs are unreadable by contracts and deleted after about eighteen days — and that willingness to forget is precisely what collapsed the cost.

06

Dencun and the Fee Collapse

EIP-4844 went live as the centerpiece of the **Dencun** upgrade on March 13, 2024, a coordinated hard fork combining the Deneb changes on the consensus layer with the Cancun changes on the execution layer. Its effect on layer-2 economics was immediate and dramatic.

To understand why, recall the world before blobs. Rollups — networks that process transactions off to the side and post compressed summaries back to Ethereum — had only one place to put that data: **calldata**, the input field of a normal transaction, which every node stores permanently and which competes head-on for execution gas. Rollups were paying premium, permanent-storage prices for data they only needed to be available briefly. EIP-4844 handed them a purpose-built alternative, and they migrated almost overnight. Transaction fees on major rollups fell by roughly an order of magnitude — commonly cited as around ten times cheaper, with simple transfers dropping well below a cent. It remains one of the most tangible single-upgrade improvements in Ethereum’s history, felt directly by every user of an optimistic or zero-knowledge rollup.

07

From Launch Parameters to Today

EIP-4844 launched conservatively. At Dencun the target was 3 blobs per block and the ceiling was 6 — deliberately modest numbers, chosen so node operators could absorb the new bandwidth without strain while the mechanism proved itself in production. Those figures were always meant to rise as confidence grew.

They have. The Pectra upgrade on May 7, 2025, via EIP-7691, doubled the settings to a target of 6 blobs and a maximum of 9. Then the Fusaka upgrade around December 3, 2025 shipped PeerDAS (EIP-7594), the first real deployment of the data-availability sampling that proto-danksharding had only prepared for — letting validators verify blob availability by checking small samples rather than downloading everything. A series of lightweight blob-parameter-only forks in late 2025 and early 2026 pushed the target and maximum higher still. Each step tightens the gap between what shipped in 2024 and the full vision, without ever re-opening the transaction format EIP-4844 established.

08

Why Proto, and Not Full

The “proto” in proto-danksharding is an honest label. EIP-4844 shipped the format, the KZG cryptography, the precompile, and the independent fee market — but not the sampling that would let each node download only a fraction of the data. At launch, every node still downloaded every blob. That is affordable at 3 to 6 blobs a block; it does not scale to the roughly 64 blobs and 16 MB per slot that full danksharding envisions.

Full danksharding closes that gap with two-dimensional erasure coding and full data-availability sampling, and as of mid-2026 it is still a future milestone, approached in careful increments rather than a single leap. EIP-4844 was the foundation stone: unglamorous, precisely specified, and load-bearing. Everything since — Pectra’s higher limits, Fusaka’s sampling, the steady march toward full danksharding — builds on the transaction type and the commitments this one proposal put in place. For the wider arc of how these blobs fit Ethereum’s scaling thesis, our report on [what danksharding is](#) carries the story forward from here.

“Let all things be done decently and in order.”

1 CORINTHIANS 14:40

METHODOLOGY & SOURCES

This report was compiled from the official EIP-4844 specification and corroborating primary and secondary sources, cross-checked by a multi-agent research review. Technical parameters — blob size (4,096 field elements × 32 bytes = 131,072 bytes), transaction type 0x03, the BLOBHASH opcode (0x49), the point-evaluation precompile at 0x0A (50,000 gas), and the versioned-hash construction (0x01 + last 31 bytes of SHA-256 of the KZG commitment) — follow the specification directly. Launch parameters (3 target / 6 max blobs, ~18-day retention across ~4,096 epochs, 1-wei blob base-fee

floor) and the Dencun activation date (March 13, 2024) are verified. Roadmap milestones — Pectra (May 7, 2025, EIP-7691, 6/9 blobs), PeerDAS in Fusaka (~December 3, 2025, EIP-7594), and subsequent blob-parameter forks — are stated directionally as of mid-2026; full danksharding remains a future milestone. Layer-2 fee reductions (~10×) are directional and vary by rollup and congestion. The KZG trusted-setup ceremony (“Summoning the Setup,” ~141,000+ contributions, 1-of-N honesty assumption) is noted as verified. A third-party bridge/restaking incident sometimes circulated as a 2026 Ethereum event was reviewed and excluded as unrelated to EIP-4844 or the blob mechanism. See eips.ethereum.org/EIPS/eip-4844, ethereum.org/roadmap/dencun, and ethereum.org/roadmap/danksharding. Educational only; not financial advice.

Alain AI Lab — independent crypto & AI research.

intelligencecrypto.org · This document is for educational purposes only and is not financial advice.

© 2026 Alain AI Lab. All rights reserved.