BTX Changes Its Proof-of-Work at Block 185,000: What MatMul v4.7 Actually Does

A hard fork at a fixed height turns mining from a lottery into a computation. Here is what the code says, what we measured, and what it costs.

AUGUST 9, 2026 · 11 MIN READ · EASYBTX.COM/RESEARCH/MATMUL-V4-7-FORK

ABSTRACT

At block 185,000, on 10 August 2026, BTX replaced its mining algorithm with MatMul v4.7. One attempt went from microseconds to roughly 30 seconds on reference hardware. We read the consensus source, ran the fork on a private test chain, and measured what changed for miners. Written the day before activation, with a dated postscript on how it landed.

Postscript added 22 August 2026
This article was published on 9 August 2026, one day before the fork. Not a word of the article below this box has been rewritten. A forecast is only worth reading if you can see how it landed, so the predictions stand as written and the record goes here. Answers in the FAQ at the foot of the page have been corrected in place, each substantive correction carrying its own dated note.

The fork activated on schedule. Block 185,000 arrived on 10 August 2026 and MatMul v4.7 took effect. The corrective pull request described further down was never merged and no corrective release was published, so the article's call on that was right.

Macs could not mine for nine days. The new work needs an ENC-RC solver and the Metal one did not exist at activation. easyBTX 0.20.0, on 19 August 2026, was the first Mac build that could mine the post-fork chain.

Then the network wedged. On 17 August 2026 the chain stalled at height 191,690. BTX Core v0.33.3, published the same day, introduced a difficulty floor that activates at block 191,714, and the chain recovered. Nodes still on older builds park at 191,713 and go no further. That upgrade wave is largely over: of 58 community nodes, 29 run v0.33.3.

The chain passed block 197,000 on 22 August 2026.

Mining is now counted in episodes, not nonces. Before the fork one attempt was one nonce, and rates were quoted in millions of nonces per second. One attempt is now a single ENC-RC inference episode, so the readout moved to episodes per minute in easyBTX 0.21.1. The old unit made a perfectly healthy Mac display a rate of zero, which is why the change was made.

Two things in this article have not held up, and we are not going to repair them quietly.

The GPU floor. The article names Ampere-or-newer as the NVIDIA requirement, following the MATADOR engine's own release note. We can no longer stand behind any specific minimum generation. The sources we can check disagree with each other and we have not settled the question, so we are calling it unresolved rather than repeating a number. What we will say is that Linux and Windows mining needs an NVIDIA GPU, and that the application tells you if your card cannot run the current engine.

The Mac gate. The article's reading of the consensus source held up. There is no chip allowlist in the mining path, and the gate really is byte-exact self qualification. Anywhere you have seen easyBTX say a Mac needs an M5 or newer, that claim was wrong and we retracted it on 20 August 2026. The requirement is any Apple-silicon Mac on macOS 26.4 or later whose GPU passes a half-second self-qualification check. Older qualifying Macs mine correctly, they just mine slower.

The version guidance is superseded. This article told easyBTX users to be on v0.13.0. Current builds are 0.21.1 on Mac and 0.22.0 on Linux and WSL. Those two lines alternate minor version numbers and are separate products, so neither is newer than the other.

What we wrote afterwards: The week the chain froze, and the day it came back, Why nodes get stuck, and how to tell if yours is, The day BTX halted, a MacBook kept the chain's memory, and Mining BTX on a Mac: 30 shares, two pools, and a 25x difficulty week.

At block 185,000, BTX replaces its proof-of-work. This is not a difficulty adjustment or a parameter change. The computation miners perform to produce a block becomes a different kind of computation, and the change arrives at a fixed height with no vote, no signalling period and no grace window.

We read the consensus source of the released v0.33.2 node, ran the activation on a private test chain to observe it directly, and measured the new engine on real hardware. This article reports what we found, including the parts that are uncomfortable.

What mining BTX looks like today

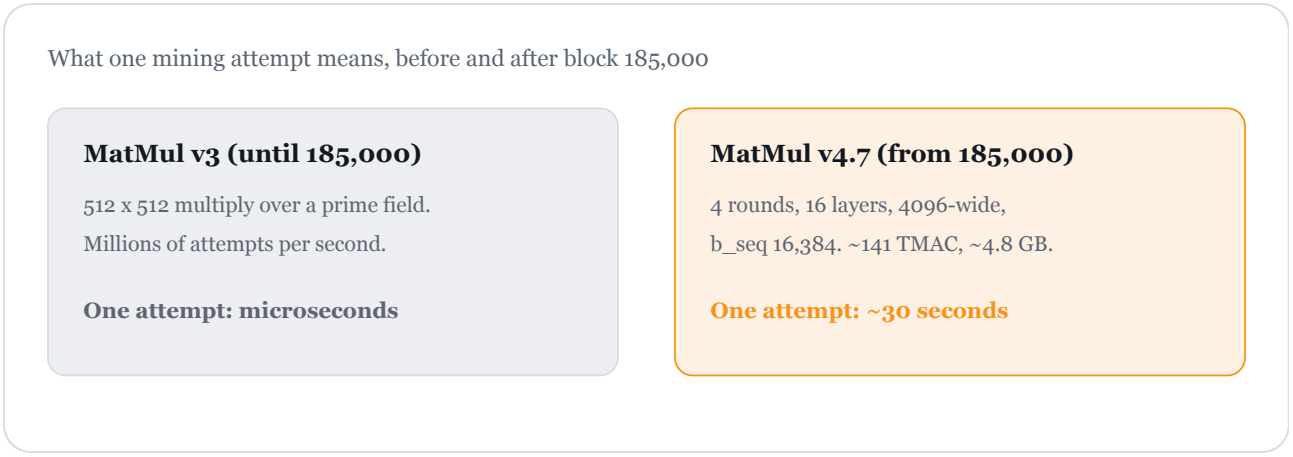
Proof-of-work is normally a lottery. A miner picks a number, runs a cheap calculation on it, checks whether the result clears a target, and discards it if not. Then it repeats, millions of times per second. The calculation has to be cheap, because the security model depends on drawing an enormous number of tickets.

BTX has always been slightly unusual here. Its v3 work performs a genuine 512 by 512 matrix multiplication over a prime field, which is far heavier than a plain hash. Structurally, though, it remains a lottery. Attempts are cheap and there are millions of them.

What replaces it

MatMul v4.7 Profile 1, named the Resident Curriculum in the source, turns each attempt into a substantial computation.

A single attempt runs a full deterministic episode: four sequential rounds through sixteen feed-forward layers, on matrices 4,096 wide, with a sequence dimension of 16,384. That amounts to roughly 141 trillion multiply-accumulate operations in exact integer arithmetic, holding about 4.8 GB of state resident on the accelerator throughout.



The workload does not get slightly heavier. It changes category. Figures from the BTX v0.33.2 MatMul v4.7 transition roadmap and consensus source.

The project's published measurements make the scale concrete. Over 100 distinct block headers, with zero CPU fallbacks, a single attempt takes 28.2 seconds at p99 on an Apple M4 Max using the Metal backend, and 32.7 seconds at p99 on a Blackwell-class NVIDIA card with 16 GiB using CUDA.

On some of the fastest silicon available, one attempt takes about half a minute. Mining stops being a question of how many guesses per second and becomes a question of how quickly a machine can finish one large exact computation.

Why a chain would do this

Cheap and fast is precisely what purpose-built hardware excels at. A lottery with very cheap tickets is an open invitation to build a machine that does nothing but buy tickets quickly, which is the road most proof-of-work chains have travelled. It ends with mining concentrated in a small number of industrial farms.

A heavy, exact, sequential and memory-hungry workload is harder to shortcut. The work is dominated by large dense integer matrix arithmetic that already maps efficiently onto tensor cores, so there is less headroom for a specialised chip to do something clever. Steps cannot be skipped either, because verification replays the episode exactly and compares digests.

The design documents name a second goal directly: pricing the older, cheaper hardware tail out of rental viability. Rented consumer GPUs are a standard attack surface for small chains, because an attacker can hire a large quantity briefly and cheaply. Raising the cost per attempt raises the cost of that attack.

The trade-off is equally direct, and worth stating rather than selling around. It raises the floor for everyone, including honest small miners. Cards that mine BTX profitably today will not mine it at all tomorrow.

What does not change

The block header stays exactly as it is at 182 bytes, and this stage adds no proof data to blocks. Wallets, balances, addresses, payouts, transaction history and the explorer are untouched.

That has a practical consequence beyond reassurance. Because the header and the job format are unchanged, the pool protocol survives intact. Pools hand out work the same way they always have. What changes is what the miner does with that work.

We ran the fork before it happened

BTX's testnet and signet both keep the activation disabled, so there is no public network on which to rehearse this. The only place the transition can be observed before it reaches mainnet is a private regtest chain, where the same code activates at a much lower height.

We ran it. Mining past the activation point on a local v0.33.2 node and capturing the work template on both sides produced the clearest single piece of evidence about how the switch presents itself:

Block	encoding_profile	matmul dimension
Before activation	`ENC-BMX4C-LT`	256
After activation	`ENC-RC`	256

The profile identifier changes. The matrix dimension does not.

That detail matters for anyone writing mining software. The obvious way to detect the fork is to watch for the workload getting larger, and on mainnet the dimension does move from 512 to 4096. But the field the node actually switches is `encoding_profile`, and a detector keyed only on dimension would have missed a pool that kept the dimension the same. We had written exactly that detector, and the test chain corrected it before it shipped.

The same capture confirmed that the proof-data budget stays at zero across the boundary, which is what the roadmap promises for this stage.

What miners will see at the moment it happens

Displayed difficulty collapses. At activation the network difficulty drops by a factor of roughly 120,000, because a unit of work now means something enormously larger than it did one block earlier. The retarget algorithm has a one-hour half-life and settles this over about 40 blocks. It is expected behaviour, not a bug and not an attack.

Blocks may arrive quickly for the first hour. Same cause, same resolution. The difficulty algorithm needs a little time to find the new level.

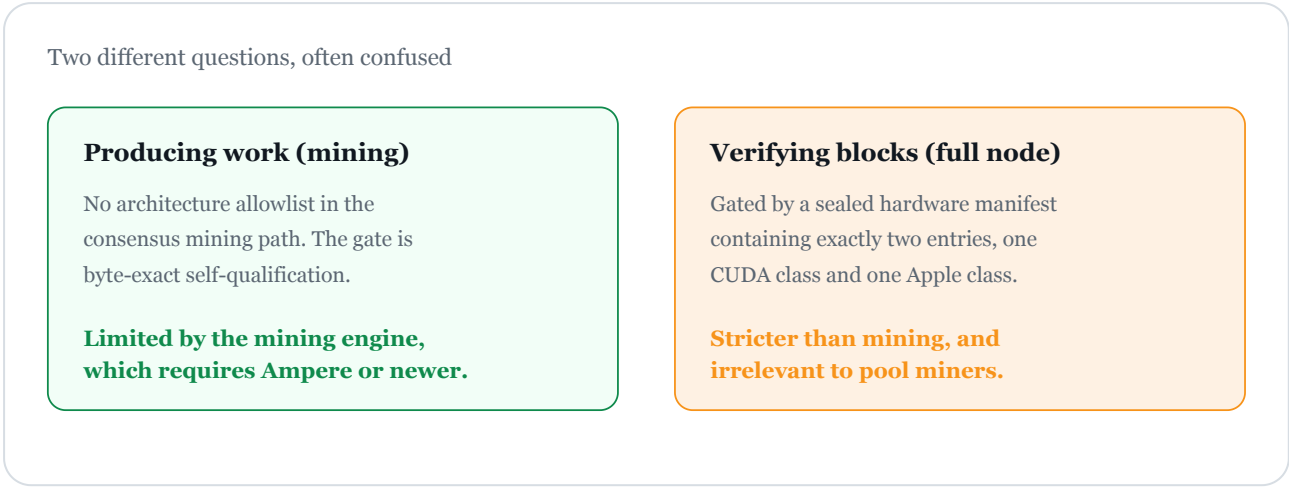
Pools may go quiet just before the boundary. One pool operator has said they may pause issuing old-style work shortly before 185,000, so that no job straddles the activation. A brief silence immediately before the switch is the pool being careful rather than failing.

Incompatible miners are refused rather than fooled. The pools have confirmed fail-closed behaviour: a miner that cannot perform the new work receives a `notify_incompatible` message and is disconnected. It is not permitted to connect and display a healthy hashrate while earning nothing. This is a good decision. Silent failure is the worst outcome in mining, because the operator keeps paying for electricity while the dashboard looks fine.

Hardware, honestly

The new work leans heavily on integer tensor-core operations. The MATADOR engine, which most easyBTX users run, has retired its Pascal, Volta and Turing builds and states that the new proof-of-work requires Ampere-or-newer tensor cores. In practice a GTX 10-series, GTX 16-series or RTX 2070 can mine right up to block 185,000, and we expect it to stop there.

We want to be precise about the boundary of what is actually known, because this affects people's hardware decisions.



The restriction most miners will hit comes from the mining engine, not from the chain. The stricter hardware list in the node governs replay-verification of blocks, which pool miners never perform.

Reading the consensus source, the mining path contains no architecture allowlist. The gate on producing work is byte-exact self-qualification: a device either reproduces the expected result or it does not. The stricter hardware list that exists in the node governs verifying blocks by replay, which is a different activity that pool miners never perform.

So the practical restriction comes from the engine authors rather than from the chain. If a new build appears, or those cards turn out to work after all, the situation could change. Plan for them stopping, but know where the limit actually comes from.

One open question worth recording

Everything above describes what the released code does. There is a caveat that belongs in an honest account.

The published v0.33.2 release carries the activation height as a compiled-in constant, and that is what nodes and pools are running. At the same time, an unmerged corrective pull request sits open

in the BTX repository which would disable the activation entirely by setting the relevant heights back to infinity. Its author describes that branch as a NO-GO for this activation, because the evidence campaign intended to authorise the fork is incomplete: the Apple half of the required hardware cohort has been reproduced, the CUDA half has not, and the project's own comparator reports that activation is not authorised.

That pull request has not been merged. No corrective release has been published. The pools are armed for 185,000. The fork will proceed.

It is still worth recording that the change is arriving slightly ahead of the project's own evidence process, and that this is visible in public in their issue tracker. Readers making decisions about hardware or capital deserve to know that, and it costs nothing to say.

What easyBTX did about it

easyBTX v0.13.0 shipped ahead of the fork. Four things in it are worth describing, because they generalise beyond this event.

It downloads and verifies its own mining engine rather than bundling one. The fork-capable build is too large to ship inside the application and moves faster than we cut releases, so it is fetched on first use and checked against the publisher's SHA-256 checksum before it is permitted to run.

It reports honestly when mining is not earning. Previously, an engine that kept running but stopped earning left the dashboard showing a confident hashrate and a share counter that quietly stopped moving. The application now watches for that specific shape, a live scan rate with no credited shares behind it, and says so.

It fails over between pools. If a pool stops accepting work, mining moves to the other verified pool within seconds. This was not a theoretical concern. During development one pool began refusing our engine while remaining perfectly healthy for its own clients, and affected users mined into nothing until they changed a setting manually.

It counts down to the fork and then confirms the outcome, reading the workload identifier from the pool's own job rather than trusting anything the application assumes about itself.

What comes after

This activation is the first of four planned stages, and only the first has a height.

The stage activating now uses the new workload with the network verifying results by replaying them exactly. A later stage would add succinct cryptographic proofs alongside that replay. A stage after that would make the proof authoritative, so that verifying a block no longer requires redoing the work, which is the point at which the design becomes genuinely elegant: expensive to produce, cheap to check. A final stage would move to a second, substantially heavier workload.

None of those later stages have activation heights. Each requires its own explicit change to the consensus rules. Nothing beyond the first stage happens at block 185,000.

Summary

At block 185,000, BTX stops being a guessing game and becomes a computation. One attempt moves from microseconds to tens of seconds. Displayed difficulty will appear to fall off a cliff, which is expected and resolves within the hour. Wallets and balances are unaffected.

Miners running easyBTX should be on v0.13.0. Miners on cards older than an RTX 30-series should expect this to be the last night those cards mine BTX.

Sources: the BTX v0.33.2 consensus source, its published measurement evidence, the MatMul v4.7 transition roadmap, a local regtest activation run performed for this article, btxchain/btx pull request #103, statements from the Byron Bay pool operator, and live network data on 9 August 2026.

Frequently asked questions

What changed at block 185,000?

Update, 22 August 2026: this happened. The fork activated at block 185,000 on 10 August 2026 and the description below is now history rather than forecast. BTX replaced its proof-of-work algorithm. Mining moved from MatMul v3, a 512 by 512 matrix multiplication repeated millions of times per second, to MatMul v4.7 Profile 1, a single large deterministic computation that takes roughly 30 seconds per attempt on reference hardware.

Is there a vote or a signalling period?

No. The activation height is a compiled-in constant in the released v0.33.2 node. The block before 185,000 runs the old rules and the block after runs the new ones. There is no grace window and no fallback to the old algorithm.

Does this affect my BTX balance or my wallet?

No. The block header is unchanged at 182 bytes and this stage adds no proof data to blocks. Wallets, balances, addresses, payouts, transaction history and the block explorer are all unaffected.

Why did network difficulty appear to collapse?

At activation the displayed difficulty dropped by a factor of roughly 120,000, because a unit of work suddenly meant something vastly larger than it did one block earlier. This was a deliberate one-time recalibration and the retarget algorithm settled it over about 40 blocks.

What GPU do I need after the fork?

On Linux and Windows you need an NVIDIA GPU. Correction added 22 August 2026: we are no longer naming a minimum generation. This article said Ampere-or-newer, following the MATADOR engine's own release note that the new proof-of-work requires Ampere-or-newer tensor cores. The sources we can check now disagree with each other about where the real floor sits, and we have not settled it, so we are treating it as unresolved rather than repeating a number we cannot stand behind. Install easyBTX and it tells you whether your card can run the current engine. On the Mac side the

requirement is any Apple-silicon Mac on macOS 26.4 or later whose GPU passes a half-second self-qualification check.

Can a GTX 1080 or RTX 2070 still mine BTX?

Until block 185,000, yes. After it, we expected them to stop, because the restriction comes from the mining engine authors rather than from a rule written into the chain. Update, 22 August 2026: we can no longer confirm where that hardware floor actually sits, because our sources disagree, so we are not going to state one. Run easyBTX and let it answer for your specific card.

What do easyBTX users need to do?

Update, 22 August 2026: v0.13.0 is superseded. Run the current build, which is 0.21.1 on Mac and 0.22.0 on Linux and WSL. Those two lines alternate minor version numbers and are separate products, so neither is newer than the other. The capabilities described here shipped and remain: easyBTX downloads and verifies a fork-capable mining engine, fails over between pools automatically, and reports honestly if the engine is running but not earning.

Why would a chain make mining more expensive per attempt?

To resist specialised hardware. A lottery with very cheap attempts rewards whoever can build a machine that makes attempts fastest, which historically concentrates mining in industrial ASIC farms. A heavy, exact, sequential and memory-hungry workload is harder to shortcut, and it also raises the cost of renting consumer GPUs to attack a small chain.

Is the mined work useful to anyone outside the chain?

Not directly. As with MatMul v3, the inputs are derived deterministically from the block and only a digest is retained. What the design changes is the hardware the network attracts and the cost of producing a valid block.

What happens to a miner that cannot do the new work?

The pools have confirmed fail-closed behaviour. An incompatible miner receives a notify_incompatible message and is disconnected, rather than being allowed to connect and display a healthy hashrate while earning nothing.
