

THE HIDDEN ECONOMY INSIDE EVERY BLOCK



By **bitcoincashautist** | 2026

THE HIDDEN ECONOMY INSIDE EVERY BLOCK

When you watch a new Bitcoin Cash block appear on a block explorer, what are you actually seeing? Most people would say: "Miners created new coins." But this common understanding has it backwards, and correcting this misconception reveals something remarkable hiding in plain sight within every block header.

The Blockchain as Buyer

Here's what's actually happening: the network itself mints currency from thin air per the block reward schedule, accumulates transaction fees (from mempool transactions), and then offers it all as payment. Payment for what? For hashes, the computational work that secures the chain by increasing the "cost to undo".

Think of it this way: miners aren't creating money. They're selling a commodity (hashes) to a buyer (the network). The network pays with newly minted coins plus fees. This perspective reveals some interesting practical implications.

Every blockchain with a Difficulty Adjustment Algorithm (DAA) has a built-in market maker. When miners flood the market with too many hashes, blocks come too fast, and the DAA responds by raising the difficulty, effectively lowering the BCH paid per hash. When miners withdraw their hashpower, blocks slow down, and the DAA lowers difficulty, raising the BCH paid per hash to attract sellers back.

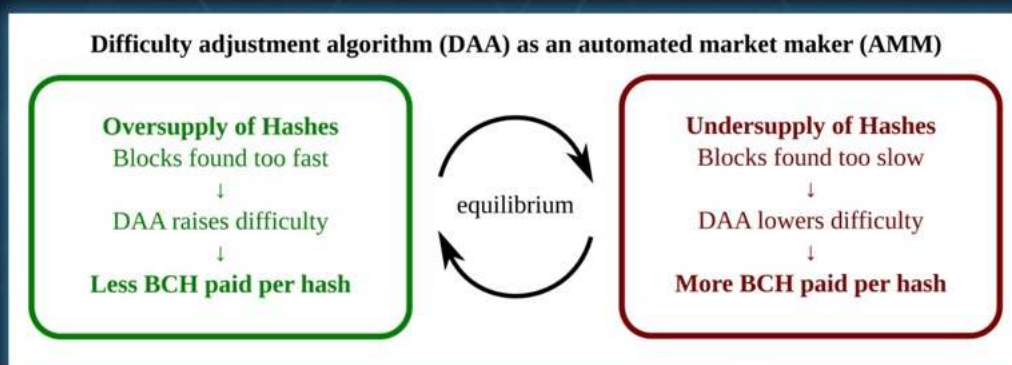


Figure 1. Difficulty adjustment Algorithm (DAA) as an automated market maker (AMM)

As shown in **Figure 1**, this is precisely how an Automated Market Maker (AMM) works in DeFi: an algorithm that automatically adjusts prices based on supply and demand, without any human intervention or external price feeds.

Every Block Is a Price Oracle

If the DAA functions as an AMM facilitating trades between the network and miners, then every block records a trade. And every trade has a price. We can calculate this price directly from data already embedded in block headers:

$$\text{price} = \text{difficulty} / (\text{reward} + \text{fees})$$

The difficulty tells us how many hashes were statistically required to find the block. The reward plus fees tells us what the network paid. Divide one by the other, and you have the exchange rate. Note that a difficulty of 1.0 corresponds to 4.295 gigahashes (GH), which we can then easily convert to other common units like exahashes (1 EH = 1 billion GH).

This creates something unique in the oracle landscape. Traditional price oracles rely on external data: someone must attest that the BCHUSD price is such-and-such, and you must trust that attestation. But the BCHEH price requires no external data whatsoever. It's computed entirely from information already validated by every full node. It's as decentralized as the blockchain because it's intrinsic to the blockchain itself. Plotting these prices reveals how the hash-as-commodity market has evolved.

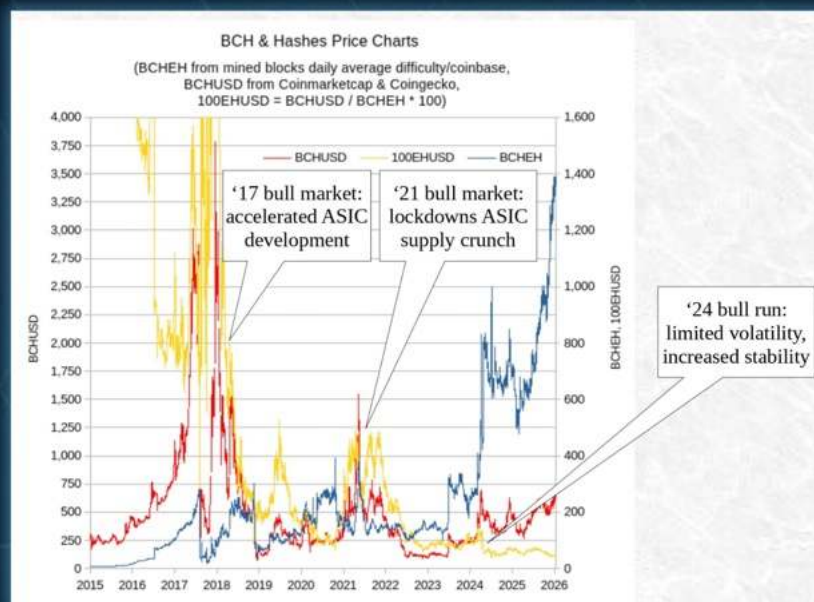


Figure 2. BCH & Hashes Price Charts

On **Figure 2**, notice the sharp drop in hash prices (yellow line) during the '17 bull run: increased demand for hashes from rising BTC and BCH prices accelerated ASIC development, which rapidly reduced the energy cost of "extracting" this digital commodity.

During the '21 bull run we see the opposite: roughly a 5x spike in hash prices. What happened? A supply crunch. Efficient ASIC technology existed, but manufacturers couldn't bring units to market fast enough to satisfy demand. This was likely a consequence of COVID-19 lockdowns disrupting global semiconductor supply chains, similar to volatility observed in other commodities during the same period.

The '24 bull run shows a maturing market. Some volatility occurred, but far less than previous cycles. As ASIC manufacturing and mining operations continue to mature, we expect this trend toward stability to continue.

From Hashes to Energy

Hashes are abstract, but they're tethered to physical reality through an inescapable constraint: producing a hash requires energy. No exceptions, no workarounds. This is enforced by thermodynamics, not protocol rules. Energy is nature's own native currency, and nobody can cheat nature's energy accounting.

If we can model the energy efficiency of mining hardware over time, we can transform our BCHEH oracle into something more tangible: a BCHMWH oracle.

Mining hardware efficiency doesn't improve linearly or exponentially forever. Like many technological processes, it follows a logistic curve with rapid early gains that gradually slow as physical limits approach, the classic S-curve of maturing technology:

$$\text{asic_efficiency} = L / (1 + e^{(-k * (x - x_0))}) ,$$

where:

- x is the input time,
- L is the asymptote (the ceiling that asic_efficiency approaches but cannot exceed),
- k determines how steep the growth curve is,
- and x₀ anchors the curve to a known timeline.

Current 5nmASICs achieve roughly 70 GH/J. The theoretical ceiling, limited by atomic-scale constraints (silicon atoms are 0.2nm in diameter), might be around 700 GH/J, and we set this as asymptote L. To find k and x0 we gathered ASIC data from hashrateindex.com and fit two curves.

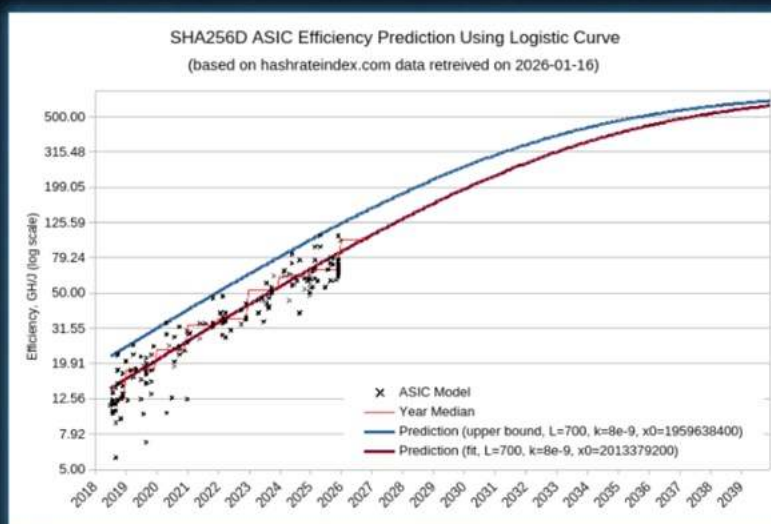


Figure 3. SHA256D ASIC Efficiency Prediction Using Logistic Curve

With this efficiency curve, each block yields an estimated mining energy price:

$$\text{BCHMWH} = \text{BCHGH} / (\text{asic_efficiency} * 3.6\text{e}9)$$

where 3.6e9 is the conversion factor from GH/J to GH/MWh. By plugging this in we can generate a price chart from blockchain records.

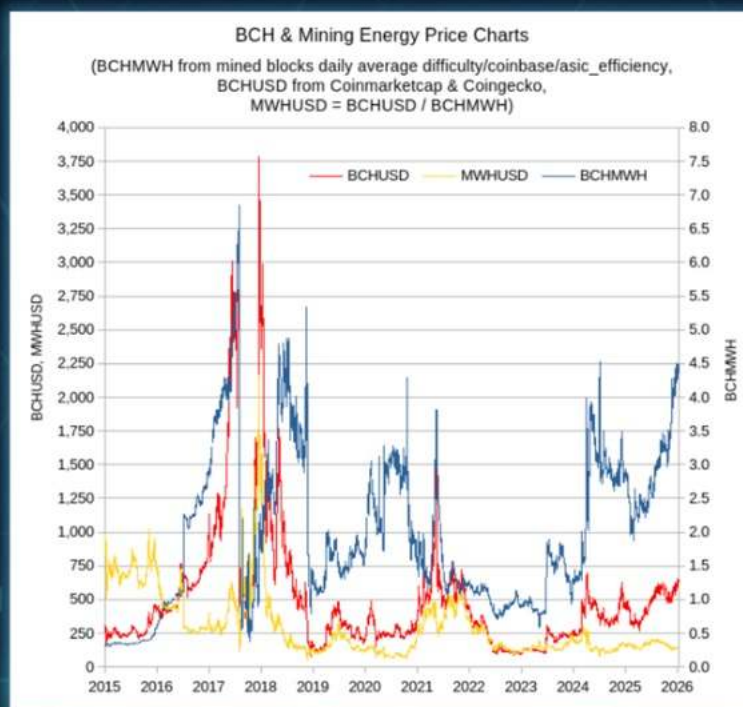


Figure 4. BCH & Mining Energy Price Charts

The model isn't perfect in **Figure 4**. It can't account for regional electricity prices, the age distribution of deployed hardware, supply crunches, or operational overhead. But from 2018 onward, as the mining industry matured, the derived price has been stabilizing. The MWHUSD chart shows remarkable steadiness even during volatile BCHMWH moves in the 2023-2025 period.

Automatic Minimum Fee Adjustment

The minimum relay fee is one of those parameters that nobody wants to think about until it becomes a problem. Set it too high during a price surge, and ordinary transactions become expensive. Set it too low, and the network becomes vulnerable to spam attacks. Historically, adjusting this parameter has required social coordination, which always lags behind market conditions.

What if the minimum fee could adjust itself?

The idea of automated fee adjustment isn't new, but previous discussions always stumbled on the same obstacle: where does the price data come from? Any external oracle introduces external dependencies, trust assumptions, and attack vectors. A malicious or compromised oracle could manipulate fees to either price out legitimate users or open the door to spam. A legitimate oracle could simply become unavailable.

The difficulty-derived price oracle solves this. Since the BCHMWH price is computed entirely from block headers that every node already validates and has available, we can express the minimum fee not in satoshis, but in energy units.

Consider a minimum fee defined as: "A typical 200-byte transaction must offer at least 10 watt-hours as fee." The satoshi amount would float automatically based on the chain's own view of energy costs. No coordination required, no external dependencies, no trust assumptions beyond those already inherent to proof-of-work.

See the chart in the next page for how this would look in practice.



Figure 5. Automatic Minimum Fee Using Native MWH Oracle

There would still be some volatility during big market swings, but much less than if defining the fee in satoshis. Notice how in '24 it barely did a ~3x move from '23 floor (while BCHUSD did ~7x), and how it quickly fell back to ~0.001 USD (while BCHUSD stayed at ~3x the '23 floor).

A backwards-compatible approach would use the all-time-high BCHMWH price rather than the current price, to avoid breaking hard-coded-fee smart contracts by making them unspendable. This creates a one-way ratchet: fees can decrease when new price highs are reached, but never increase due to price drops.

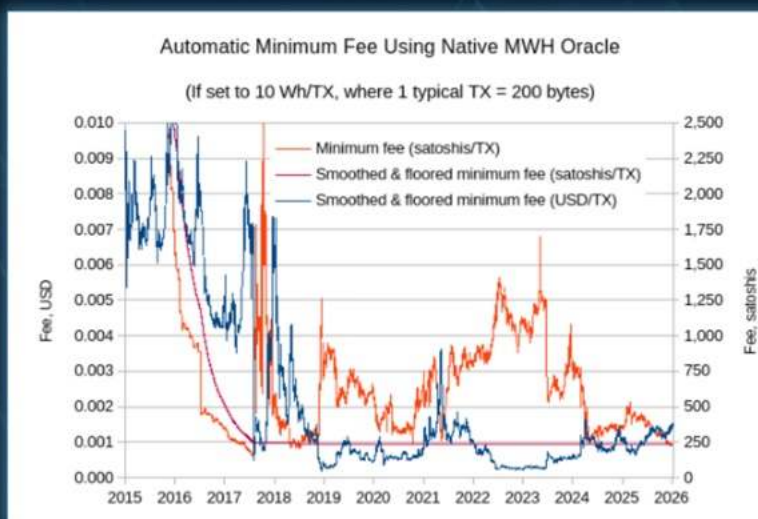


Figure 6. Automatic Minimum Fee Using Native MWH Oracle

In **Figure 6**, we can see how the fee established in '17 would hold until today, because BCH price hasn't made new all-time-highs since then. This method would result in generally cheaper minimum fee than free floating fee.

Either way, anchoring fees to energy rather than arbitrary sat values gives us a physically meaningful spam threshold. An attacker must always burn real-world resources proportional to the attack, regardless of BCH's exchange rate. The fee becomes a direct measure of the cost to abuse the network, denominated in the same fundamental unit that secures the chain itself.

What happens as BCH scales? If every byte is priced in a fixed amount of energy then the network's security budget grows with adoption. At different block sizes, this translates to:

Blocksize, MB	1-conf Security, MWh	51% Attack Power, MW	Yearly Mining Energy, TWh	Exceeds Country
1	0.05	0.3	2.63	Suriname (#148)
8	0.40	2.4	21.04	Iceland (#79)
32	1.60	9.6	84.15	Finland (#42)
64	3.20	19.2	168.30	Poland (#27)
128	6.40	38.4	336.61	Turkey (#15)
256	12.80	76.8	673.21	South Korea (#7)
512	25.60	153.6	1,346.43	Russia (#4)
1024	51.20	307.2	2,692.86	India (#3)

Figure 7. Energy Consumption and Attack Power Scaling by Block Size

At 32 MB blocks, the current blocksize limit, user fees would purchase the network more energy than Finland, making a sustained 51% attack require nation-state-level power generation.

Building Native Steadycoins

DeFi has produced numerous "stablecoins" pegged to external assets like USD. These require external oracles: trusted third parties who attest to off-chain prices. This introduces trust assumptions and attack surfaces.

What if we could build a steadycoin using only on-chain data? Oracle contract would parse and verify headers and coinbases and emit BCHEH price information to other contracts. This is already feasible with BCH's current smart contract capabilities, though careful design is required to work around coinbase transaction sizes and possibility of forks and reorgs. Such oracle would have SPV (simple payment verification) security which would make the price information as secure as the accumulated blockchain proof-of-work.

Then, using mechanisms similar to Liquity-style lending protocols, we could create a "ExaHashCoin" (EHC) whose value tracks the cost of computational work:

1. Users deposit BCH as collateral
2. They mint EHC against this collateral
3. The collateralization ratio is governed by the on-chain BCHEH price
4. Liquidations and redemptions maintain the peg

Taking it further: plug in the efficiency model, and you get Mega-Watt-Hour-Coin (MWHC), a token tracking the mining energy cost of hashes. This wouldn't precisely mirror global energy prices, but it would track some aggregate of mining economics, dominated by energy costs but also affected by ASIC supply chains, operational expenses, and similar factors.

What This Means

The insight that "the network buys hashes from miners" rather than "miners create coins" reveals that every proof-of-work blockchain already contains a functioning commodities market with an automated market maker, and that the market's trade history is permanently recorded in every block header.

This transforms block headers from mere structural data into a maximally decentralized oracle, one that requires no trust in external attestors because it requires no external data at all. Building financial infrastructure on this foundation means building on the same security model as the chain itself.

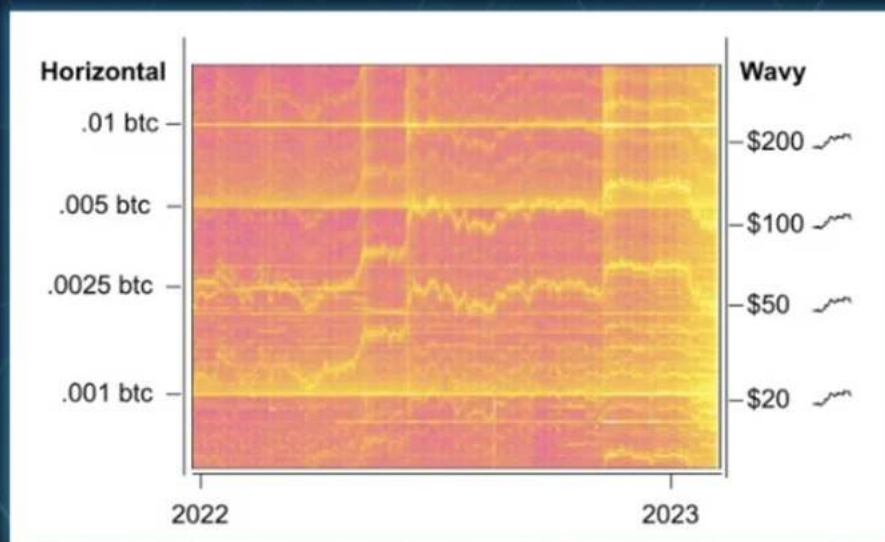
The price of an exahash can be the foundation for a new category of DeFi primitives, ones that inherit their security properties directly from proof-of-work rather than from trusted intermediaries.

Acknowledgments

This is not the first work on this idea. It was actually inspired by Karol "Licho" Trzuszczkowski's work, who, in 2021, published a paper titled "Proportional block reward as a price stabilization mechanism for peer-to-peer electronic cash system." Based on that he started the Ergon (XRG) blockchain, a code-fork of BCH with a new genesis block. The idea of an energy-tracking coin stuck with me ever since I learned about it, and to me it first revealed the necessity of predicting ASIC efficiency (the paper used pure exponential function) to make it work.

Licho's paper referenced Vitalik Buterin's 2014 post titled "The Search for a Stable Cryptocurrency", which shows us that similar ideas and desires have been afloat for more than a decade.

More recently, I have learned about UTXOracle (Daniel Hinton, Steve Jeffress, "UTXOracle: A Decentralized Approach To The Oracle Problem", 2023) which infers price of Bitcoin using a fascinating and novel method: it takes mempool samples and performs analysis on frequency of new output amounts. It reveals that, often enough, when users make payments they're sending amounts that correspond to rounded USD value (10, 50, 100). Plotting the frequency reveals strong patterns which almost perfectly match real-time market prices.



We share the same motivation: what if we could have a native price oracle, to be used as a DeFi building block? This is an exciting frontier, and solving this problem could help blockchains decouple from their dependence on fiat markets. Godspeed!

ABOUT THE AUTHOR



@bitcoincashautist

I like cash and when I say cash I mean Bitcoin Cash. Accelerate permissionless economy!"

About

Name inspired by the WSB GameStop event. I joined team BCH in '21 when I made it my mission to help L1 native tokens to BCH (being proposed as "Group tokens" at the time). Since then: I learned Script, contributed a few CHIPS and pull requests, wrote a few papers, and made a few apps. It's been a wild ride and I enjoyed every moment of it! Let's accelerate permissionless economy together!

Socials



gitlab.com/0353f40e

Support the Author

If you found this helpful or enjoyed it, you can support my work here:



bitcoincash:zzl7nm2w48rcxrpwa9kp0gaatl6k8sf09vpje347m4