EPIC CASH

TECHNICAL WHITEPAPER · 2026 EDITION

EPIC PRIVATE INTERNET CASH

A Peer-to-Peer Electronic Cash System

THE EPIC COMMUNITY

Network data as of July 2026 · explorer.epiccash.com

Mainnet since September 2, 2019

Contents

I · Abstract

Epic Private Internet Cash ("Epic") is a Layer 1 blockchain protocol — a complete, sovereign network with its own consensus, mining, and monetary policy, not a token issued on another chain — implementing a peer-to-peer electronic cash system that delivers privacy, fungibility, and scalability at the base layer. Launched on September 2, 2019 through a fair launch — no initial coin offering, no premine, no venture allocation — Epic implements Mimblewimble, a blockchain protocol that records no addresses, stores no plaintext amounts, and discards spent transaction data while preserving full verifiability. Consensus is secured by a polyphasic proof-of-work spanning three mining algorithms across three hardware classes (CPU, GPU, ASIC), under Feijoada, an extensible scheduling policy through which additional proof-of-work algorithms can be introduced by consensus upgrade; monetary policy follows Bitcoin's fixed maximum supply of 21 million coins, and governance is deliberately patterned on Bitcoin's: no company, no foundation, no protocol-controlled treasury, and no central point of control.

This document describes the Epic protocol as it operates today, after nearly seven years of mainnet operation, three major network upgrades, and sustained development by a decentralized global community. Where the original 2019 whitepaper described intentions, this edition describes results: a complete blockchain of approximately 4 GB after more than three million blocks, transactions used as a payment rail on multiple continents, and a network whose security model has withstood the failure of the exchanges, intermediaries, and market structures around it.

The original design thesis has not changed. Money that can be surveilled can be controlled — and so can money that depends on centralized entities and institutional, regulatory-dependent infrastructure in order to move at all. Money that can be controlled ceases to be the property of its holder. Epic exists to provide the alternative: unconditional money — private by default, fungible by construction, mineable on ordinary hardware, dependent on no institution, and governed by no one.

This document concerns Epic itself. A companion paper, Privacy-Currency Architectures: Epic, Monero, Zcash, provides detailed comparative analysis of the major privacy networks.

II · Design Goals

Epic was designed to restore the properties that made Satoshi Nakamoto's original proposal revolutionary, incorporating a decade of lessons from Bitcoin's evolution. Four properties define the system, and every protocol decision serves at least one of them.

Private by default. Privacy in Epic is not an optional feature, a second transaction type, or an opt-in shielded pool. It is the only mode of operation. The ledger records no addresses and no plaintext amounts; transaction data is aggregated and pruned as a matter of protocol. Systems that make privacy optional make it suspicious — when only a minority of transactions are shielded, the act of shielding is itself a signal. Epic's uniform privacy means no transaction stands out, because no transaction reveals more than any other.

Fungible by construction. Fungibility is the property that makes every unit of a currency interchangeable with every other — the reason one dollar bill is as good as any other dollar bill. Because Epic's ledger retains no per-coin history, no coin can be distinguished from any other. There are no "tainted" coins, no blacklists, and no two-tier market between fresh and circulated units. Fungibility is not a policy Epic enforces; it is a property the data structure makes unavoidable. The growth of chain-surveillance infrastructure has made this failure mode concrete for transparent blockchains, where coins routinely trade at different effective values depending on their history.

Scarce and predictable. Epic follows Bitcoin's monetary DNA: a hard cap of 21 million units, with the overwhelming majority of supply emitted in the network's first nine years. Epic's circulating supply synchronizes with Bitcoin's at the Singularity in May 2028, after which the two currencies share an essentially identical scarcity profile. One Epic is divisible into 100,000,000 freeman, mirroring Bitcoin's satoshi structure.

Decentralized and capture-resistant. Epic is mineable on the hardware people already own — CPUs and GPUs numbering in the billions worldwide — through a three-algorithm proof-of-work design that resists consolidation by any single hardware class. Its governance is the Bitcoin model: open-source software that node operators freely choose to run, with no foundation, corporation, or on-chain voting structure to capture, coerce, or subpoena. The network's history, described in the following section, is the direct product of this design.

III · Seven Years of Mainnet

A whitepaper written before launch can only make claims. This edition can report results.

Operational record. Since genesis on September 2, 2019, the Epic mainnet has operated with a single interruption: a pause of several weeks in 2021 while the value-overflow incident described below was remediated and the chain rolled back. Outside that window, block production has run uninterrupted — through three major network upgrades, multiple market cycles, and the disappearance of most of the commercial infrastructure that once surrounded it. The network requires no company to operate, and none has ever operated it. Nodes, miners, and wallets are run by independent participants on every inhabited continent, with node connections observed from more than 130 countries as of July 2026, measured by geolocated node IP addresses on a unique-IP basis — a proxy for geographic reach, not a count of users.

Scalability proven, not projected. The 2019 whitepaper projected that a Mimblewimble chain would remain radically smaller than a transparent chain of equivalent age. Seven years later the result is empirical: after more than three million blocks, the complete Epic blockchain — full archival history — occupies approximately 4 GB, and a default pruned node stores less still. A like-for-like note, since "chain size" hides different measurements: Epic's full archival history is approximately 4 GB; Bitcoin's is roughly 750 GB, and a pruned Bitcoin node — after downloading and verifying all of it — can then run on roughly 7–10 GB of disk per Bitcoin Core documentation [27]. Epic's default pruned footprint is bounded above by the ~4 GB archival figure; precise sync-download and pruned-disk measurements have not yet been published; on storage, this paper prints only measured figures.

A pruned Bitcoin node can run on modest disk — but only after downloading and verifying three-quarters of a terabyte, and the archival record it depends on remains nearly two hundred times larger than Epic's entire history. One asymmetry cuts the other way and is stated in Section XII: Epic's default sync verifies state fully but skips historical header proof-of-work outside a recent window, whereas Bitcoin's default replays full validation; Epic's full header-PoW-validation mode is available by configuration. This is the work of Cut-Through — Mimblewimble's mechanism for discarding spent, intermediate transaction data while preserving the ability of every node to verify the full monetary supply (described in Section IV). A full validating node — the instrument of monetary sovereignty — runs comfortably on a consumer laptop, and the chain's footprint is small enough for even low-end mobile hardware; no current mobile release integrates a node, so that is a statement of resource requirements rather than shipped software, and node performance on such devices has not been benchmarked. This preserves the decentralized validation Bitcoin's growing chain increasingly reserves for well-resourced operators. Cut-Through is no longer a promising idea; it is seven years of measured behavior.

The March 2021 incident. Between March 12 and March 20, 2021, an attacker exploited a value-overflow flaw — the same class as Bitcoin's CVE-2010-5139 [10], which in August 2010 minted 184 billion counterfeit BTC out of thin air. The response followed the same precedent as Bitcoin's, eleven years apart: patch the flaw, invalidate the exploited history by rolling back to a pre-exploit block, resume. In both networks the rollback neutralized the counterfeit supply entirely — whatever total was forged, none of it survives on the chain. The residual damage was economic: approximately 2.8 million forged EPIC reached the ViteX exchange before its trading halt, with an estimated 6.6 BTC

in trader losses there, and miners across the network lost roughly 200,000 EPIC in revenue to the pause. The remediation stopped the chain for several weeks — the network's only interruption of service and only significant security event. The flaw was fixed and the network resumed operation. Contemporaneous accounts are cited in the references [23][24]; a consolidated technical post-mortem documenting the exact rollback height, root cause, patch provenance, and verification process has not been published.

Surviving the intermediaries. Since 2019, at least six exchanges that listed Epic have ceased operations entirely — among them BiHODL, CITEX, VinDAX, ViteX, and TradeOgre — the last seized outright by Canadian federal police in September 2025, in that country's largest cryptocurrency seizure and its first dismantling of an exchange [25] — an enforcement action that illustrates this paper's thesis that intermediary-dependent access is revocable, and part of why the atomic-swap and decentralized-relay work of Section XI exists — and the community itself terminated two further listings (BitMart, Coinstore) with venues that failed to meet its standards. Bitcoin survived the collapse of Mt. Gox; Epic has survived six such collapses. The pattern illustrates a design principle rather than good fortune: a currency whose custody, transaction, and mining layers require no intermediary is not destroyed when intermediaries fail. Separately, a now-defunct independent mining pool once issued representations of Epic on the Stellar and BNB networks as payout conveniences for its clients; these were never protocol assets, are no longer supported, and should not be confused with the independent wrapped-EPIC initiative described in the Development Roadmap (Section XI).

Real usage under real pressure. Epic functions today as a payment rail in precisely the circumstances it was designed for: commerce facing censorship, debanking, and infrastructure exclusion. Community-reported use includes service providers operating under sanctions pressure, hosting businesses cut off from conventional payments, and agricultural communities in West Africa transacting outside a banking system that never reached them; these accounts are deliberately left unattributed to protect the businesses involved. These users did not adopt Epic as speculation. They adopted it because it works when the alternatives are forbidden, frozen, or absent.

IV · Protocol Architecture

Epic implements Mimblewimble, a blockchain protocol whose privacy and scalability derive from a single insight: a ledger does not need to store transaction histories to prove that no counterfeiting has occurred. It needs only to verify that inputs and outputs balance and that ownership is authorized — and both can be proven with elliptic-curve cryptography alone, without addresses, without visible amounts, and without retaining spent data.

Confidential transactions. Every output in Epic is a Pedersen commitment — a cryptographic envelope of the form $C = r \cdot G + v \cdot H$, where v is the amount, r is a private blinding factor, and G, H are fixed curve generators. The commitment is computationally binding under the discrete-logarithm assumption — no party can feasibly open it to a different value — and perfectly hiding: it reveals nothing about the amount, unconditionally. Range proofs (Bulletproofs, a compact zero-knowledge proof technique) attached to each output prove that committed values are non-negative without disclosing them, foreclosing inflation through negative-value outputs. Bulletproofs deserve emphasis for what they do not require: a trusted setup. Each proof is an argument of knowledge built from an inner-product argument over the very same elliptic-curve group as the commitments themselves, and its soundness rests on the discrete-logarithm assumption alone — the identical assumption already securing the commitments. Each proof binds to its specific commitment: the commitment enters the proof's Fiat–Shamir transcript, so a proof generated for one output can never be replayed against another, and what it demonstrates is that the committed value lies within a valid 64-bit range without revealing anything further about it. Proof size grows only logarithmically with the range (675 bytes per output), and proofs verify efficiently in batches. Where other privacy systems of the same era bootstrapped their proving machinery through multi-party ceremonies — a construction known as a trusted setup, whose compromise would have permitted invisible counterfeiting — Epic's proof system is trustless by construction: no ceremony, no structured reference string, no toxic waste. Nothing to compromise, because nothing was ever trusted. Amounts therefore never appear on the ledger in plaintext, yet every node verifies that no transaction creates money.

No on-chain addresses. Epic's ledger has no address system. Ownership of an output is knowledge of its blinding factor, and transactions are constructed interactively between sender and receiver, who jointly produce the commitments and signatures. Nothing resembling an account identifier ever touches the chain. Transport-layer identifiers do exist — Epicbox addresses, Tor onion endpoints — but they belong to the communication layer, live outside the protocol, and never appear on the ledger. Where transparent blockchains such as Bitcoin leak identity through address reuse and clustering — feeding an entire chain-surveillance industry that maps flows and scores coins — Epic's ledger contains no identifiers to cluster. The interactive transaction model and its transport layers are described in Section VII.

Schnorr signatures and the kernel excess. Mimblewimble's ownership model is completed by Schnorr signatures over what is called the kernel excess. When a transaction balances — outputs minus inputs summing to zero value — the residual of its commitments is a commitment to zero whose blinding factor is the sum of the participants' secret blinds. That residual, the kernel excess, is a public on-chain value functioning as an ad-hoc public key; its corresponding private key — the sum of the participants' secret blinds — is held by no single party and exists only jointly. The trans-

action is authorized by a Schnorr signature under it. This is where interactivity becomes cryptography rather than convention: sender and receiver each hold only their own blinding factors, so neither can produce the signature alone. Each contributes a nonce and a partial signature, and because Schnorr signatures are linear, the partial signatures simply add into one valid signature under the aggregate excess — a native multi-party signing ceremony inside every ordinary payment. The same linearity is what allows transactions to aggregate within blocks without breaking verification, allows kernels to survive Cut-Through as permanent ~100-byte attestations while everything around them is discarded, and supplies the primitive — partial and adaptor signatures — on which the multisignature and atomic-swap roadmap (Section XI) is built. There are no scripts and no addresses to sign against: knowledge of blinding factors, proven by Schnorr signature, is ownership.

Cut-Through. Mimblewimble's aggregation property allows intermediate transaction data to be removed without weakening verification: when an output created in one transaction is spent by a later one, the pair cancels mathematically and can be discarded. Applied across the chain's entire history, the permanent ledger converges toward only three components: block headers (~250 bytes each), currently unspent outputs, and transaction kernels — small (~100 byte) proofs that permanently attest each transaction occurred and balanced. This is the mechanism behind the 4 GB chain reported in Section III: the ledger forgets what it does not need, and its verification is stronger, not weaker, for it. One honest caveat belongs here: Cut-Through governs what nodes must store and verify — it does not retract copies an external observer archived while data was in flight. Its privacy value is that the permanent, canonical record every future observer must consult contains no address graph and no deterministic input-to-output mapping.

Every block is a CoinJoin. CoinJoin is a privacy technique on transparent chains in which multiple parties deliberately merge their transactions to obscure which payer paid which payee. Because Mimblewimble transactions aggregate natively, all transactions in an Epic block merge into a single indistinguishable set of inputs, outputs, and kernels, with no internal mapping between them. What CoinJoin achieves on Bitcoin through deliberate, detectable coordination — and what mixing services attempt at legal peril — occurs in Epic automatically, in every block, as a structural property. The ambiguity it provides scales with block fullness — a block containing a single transaction aggregates nothing — so aggregation is an expected-case property that strengthens with usage rather than a uniform per-block guarantee. Individual transactions do exist before aggregation and can be observed during propagation — the window that Section V's network-layer defenses address — but none of that pre-aggregation structure survives into the permanent ledger.

V · Network-Layer Privacy

Cryptographic privacy on the ledger must be matched by privacy in propagation. An observer who cannot read amounts or addresses on the chain may still attempt to link transactions to the IP addresses that first broadcast them. Epic defends this layer in depth.

Dandelion++ broadcast. When a transaction enters the network, it does not diffuse immediately. It first travels a randomized "stem" path through a sequence of individual nodes, and only then enters the "fluff" phase of broad diffusion. An adversary monitoring traffic sees the transaction surface far from its origin, substantially reducing the effectiveness of first-seen source-IP mapping under the protocol's stated adversary model. Dandelion++ is a probabilistic defense with explicit assumptions — not an anonymity guarantee against a global or ideally positioned network observer — which is precisely why Epic layers additional transports beneath it. Stem-phase transactions may also aggregate with others en route, compounding the ambiguity. Topologically, Dandelion++ runs over a sparse anonymity graph, and nodes re-randomize their stem relays every epoch — on the order of minutes — reducing the usefulness of any stable vantage a malicious "first-spy" peer might try to hold.

Tor integration and OnionStem. The v4 network upgrade (November 2025) introduced Tor process integration, onion-service support, and a new ONIONSTEM peer capability through which nodes exchange Tor onion addresses. Routing Dandelion++ stem traffic end-to-end over onion connections remains under active development: the capability ships in v4 but is not yet part of the default full-node profile. Wallet-level Tor transport, by contrast, is available today and described below.

Private transaction transport. Because Epic transactions are built interactively (Section VII), the sender-receiver communication channel is itself a privacy surface. Epic provides several protected transports: direct wallet-to-wallet communication over Tor onion services; Epicbox, a relay that holds end-to-end encrypted transaction slates so the parties need not be online simultaneously and do not learn one another's network location — the relay itself sees only ciphertext and connection metadata, and connecting to it over Tor can hide the source IP — though timing, message-size, mailbox-identifier, and usage-pattern metadata may remain; and fully offline exchange of transaction files by any channel the parties choose. A migration of the relay layer to the Nostr protocol — replacing dedicated Epicbox infrastructure with a censorship-resistant network of independent relays — is on the development roadmap (Section XI).

Threat model. Privacy claims are only meaningful against named adversaries. The following summarizes what each observer class can and cannot obtain:

ADVERSARY	WHAT THEY CAN OBSERVE	PRIMARY DEFENSE / HONEST RESIDUAL
Passive chain observer	Headers, unspent commitments, kernels — no addresses, no amounts, no deterministic input-to-output mapping; a known unspent commitment can be watched until it is spent	Ledger design (Section IV); what observation cannot yield is where value went next
Malicious peer / "first-spy"	Transaction timing and ciphertext during propagation	Dandelion++ stem routing; Tor transports; probabilistic, not absolute
Global network observer	Traffic patterns across many vantage points	Tor onion transport today; end-to-end Onion-Stem in development; residual risk acknowledged
Transaction counterparty	Full details of their own transaction	Inherent to any payment; interactivity ensures receipt is consensual
Epicbox relay operator	Encrypted slates plus connection metadata: timing, sizes, mailbox identifiers, usage patterns	End-to-end encryption; Tor removes network-origin metadata though behavioral patterns can persist; Nostr migration decentralizes the relay
Exchange (deposit/withdrawal)	Amounts and identity at its own perimeter	Inherent to custodial venues; the chain behind the perimeter still yields no history
Historical data collector	Block-level input/output/kernel sets and anything archived in flight	Cut-Through defines required node state, not universal amnesia; the canonical record still yields no deterministic input-to-output mapping
Eclipse / malicious bootstrap peers	Under the default synchronization configuration, could present a fabricated post-checkpoint header history to a newly syncing node; historical header PoW and difficulty-increment checks are skipped	Checkpoint hashes are enforced at listed heights through 2,200,000; full header-PoW validation, peer diversity, out-of-band tip verification, checkpoint refresh, and default hardening reduce bootstrap risk (Section XII)

VI · Polyphasic Proof of Work

Epic secures consensus with proof-of-work distributed across three algorithms, each targeting a different hardware class, with block production allocated 48% / 48% / 4% among them:

- **RandomX (48%)** — optimized for general-purpose CPUs, the most widely distributed computing hardware on Earth. RandomX executes randomly generated programs in a virtual machine, exploiting the strengths of CPUs (branch prediction, large caches, fast memory access) to minimize the advantage available to specialized hardware.
- **ProgPoW (48%)** — optimized for commodity GPUs, designed to utilize a graphics card's full capabilities and thereby narrow the gap a purpose-built chip could open over hardware already in hundreds of millions of gaming PCs.
- **Cuckatoo31+ (4%)** — a memory-hard graph-theoretic puzzle (Cuckoo Cycle family) open to specialized hardware, retained at a small allocation to diversify the security base without permitting ASIC dominance.

At each new height, Epic recomputes its difficulty vector from the preceding chain state: the component associated with the preceding block's algorithm is retargeted using 60 observations of that algorithm's own blocks. At the target mix, those observations span roughly 125 minutes for RandomX and ProgPoW — against Bitcoin's fixed two-week epoch — and roughly 25 hours for Cuckatoo31+ at its 4% share. Independent per-algorithm targets limit cross-algorithm difficulty contamination; deterministic scheduling means, however, that loss or withholding of the required algorithm can still impair chain liveness (Section XII). Blocks arrive on a 60-second target, giving Epic ten times Bitcoin's confirmation cadence.

The allocation is enforced deterministically by **Feijoada**, Epic's consensus policy engine: at every height, Feijoada fixes which algorithm the next block must use, drawn from the policy's proportions, so miners cannot choose algorithms opportunistically and the 48/48/4 mix is a property of the chain rather than of miner behavior. The current 48/48/4 allocation was set by the v3.4 network upgrade in June 2023, but the polyphasic framework itself is open-ended: additional algorithms can be introduced by consensus upgrade and allocations rebalanced as the hardware landscape evolves. The allocation is an instrument, not a dogma. Its purpose is fixed — preserve decentralization and maintain the broadest resilient base of potential hashpower providers, so the network stays live and functioning without reliance on ongoing dedicated capital expenditure, as single-algorithm ASIC networks require — and the parameters serve the purpose, not the other way around.

The installed base: latent supply, honestly framed. Roughly 6.6 billion CPU and GPU devices exist worldwide [36] — in laptops, desktops, workstations, and phones — purchased for computing, gaming, and productivity, not for mining. This installed base is not committed defensive hashrate, and this paper does not treat it as such: realized security is only the hashpower that mining rewards actually attract, and participation depends on profitability, electricity, software, and operator choice. What the installed base is, is latent supply with structural properties no ASIC network has: anyone can begin contributing with hardware they already own at zero incremental capital expenditure, entry is permissionless and geographically dispersed, and the fleet's existence, replacement, and improvement are funded entirely by non-mining demand. The same commodity charac-

ter carries risks this paper names rather than hides — rental markets, botnets, and rapid hashpower switching — each bounded by the polyphasic cap: any one algorithm, however sourced, earns at most its scheduled share. Bitcoin's security, by contrast, rests on single-purpose ASICs that must be specifically financed, manufactured, deployed, and perpetually replaced out of mining revenue alone — a dependency whose long-term consequences are examined in Section VIII.

Attack cost, and attack complexity. A 51% attack on a single-algorithm chain is conceptually straightforward: acquire a majority of one hardware type — one supply chain, one rental market, one control surface. Polyphasic proof-of-work diversifies hardware exposure and raises the operational complexity of attack. The exact reorganization threshold depends on the attacker's share within each algorithm, the deterministic scheduling of algorithms across blocks, per-algorithm difficulty dynamics, and the accumulated-work comparison — it is not accurately summarized as "51% of all three." What can be stated precisely: an attacker seeking reorganization power must assemble work across unrelated hardware markets; no single hashpower rental market spans all three algorithms; total capture of any one algorithm still leaves the attacker producing a minority of blocks under honest conditions — though, because scheduling is deterministic, withholding a captured algorithm's blocks can degrade liveness even without reorganization power (Section XII). The polyphasic design therefore multiplies not only the cost of attack but its operational complexity, and it diffuses centralization pressure across hardware classes rather than concentrating it in one supply chain. The attack surface is, moreover, itself a policy variable: Feijoada is an extensible framework, and additional or different algorithms can be introduced through a consensus upgrade — subject to implementation, security review, parameterization, and network adoption — allowing the coordination burden on a prospective attacker to rise in step with the value being defended. It is a lever to use judiciously, since new algorithms also fragment honest hashrate and enlarge the implementation surface; community research is exploring a candidate fourth algorithm inspired by CRYSTALS-Dilithium's lattice operations (Section XI).

VII · Transactions

The interactive model. Epic transactions are constructed jointly by sender and receiver: the sender proposes, the receiver adds their output commitment and signature, and the sender completes and broadcasts. This is a departure from the broadcast-to-address model of transparent chains, and it is deliberate. Interactivity is what allows the chain to carry no addresses at all; it means value cannot be pushed on an unwilling recipient (foreclosing dust attacks, in which surveillance actors send tiny traceable amounts to tag wallets); and receipt of funds is affirmative — a property with practical significance for parties who must demonstrate consent over what they accept.

Interactivity as a foundation. Joint construction is not merely a privacy feature: it is the native primitive on which conditional and multi-party transaction types are built. Multisignature outputs, cross-chain atomic swaps, and payment-channel constructions all reduce to parties collaboratively building a transaction — which is the only kind of transaction Epic has ever had. The multisig and atomic-swap capabilities on the development roadmap (Section XI) are therefore extensions of the existing transaction flow rather than bolt-on protocol changes, and on Epic they take the form of ordinary-looking kernels rather than visible script constructions, preserving the ledger's uniformity. Interactivity also unlocks use cases push-payment systems cannot serve. Interactive construction ensures a receiver-side participant contributes to creating the transaction and can refuse it before completion — the structural basis for organizations that must be able to decline specific inbound funds at the protocol level rather than by after-the-fact return, and for regulated contexts in which an entity may accept funds only from explicitly approved counterparties with demonstrable opt-in. Where a transferable receipt or a regulated counterparty record is required, the parties retain an explicit off-chain payment proof binding the transaction to an identified receiver — a construction Mimblewimble wallets already provide in early form — since the aggregate on-chain kernel, by design, identifies no one by itself.

Transport. Sender and receiver can complete a transaction over any channel: directly via Tor onion services, asynchronously through the encrypted Epicbox relay (no simultaneous online requirement, no disclosure of either party's location), or by exchanging transaction files through any medium — including entirely offline. Transaction data can even be encoded as a string of emoji, so a complete, valid Epic transaction can travel through any messaging app, social platform, or handwritten note. Ongoing work targets transaction-reliability improvements across these methods, and the planned Nostr migration decentralizes the relay layer itself.

Fees and confirmation. With 60-second blocks and Cut-Through keeping validation light, transactions confirm quickly. Fees follow the reference node's default relay policy of 0.001 EPIC per weight unit — local, operator-configurable policy rather than a consensus rule — putting a typical transaction's default threshold at 0.047 EPIC — approximately one US cent at a representative July 2026 market price, a fiat value that is illustrative and varies with market price and liquidity. Epic's fee market has been quiet in practice, an observed condition that follows from block space remaining abundant relative to demand — the condition Epic's parameters and scaling path are designed to preserve. Bitcoin's fees, by contrast, spiked to tens of dollars per transaction at the 2017 and 2021 peaks — with peak readings as high as $87 (December 2017) and $64 (April 2021) [28], depending on tracker and averaging window — daily-average series print somewhat lower — episodes in which small payments, the defining use case of cash, were priced off the network entirely.

Layer-1 scalability. Epic's current consensus parameters cap block weight at 40,000 units; at the representative two-input, two-output transaction (weight 47), that is approximately 851 transactions per 60-second block — about 1.2 million transactions per day of theoretical capacity, roughly 14 per second sustained, orders of magnitude above present usage. The deeper point is architectural. Traditional blockchains scale with their history: every transaction ever made is retained, so a decade of payments weighs on every future node regardless of whether any of it still matters. Mimblewimble scales primarily with its current state: as coins circulate, intermediate transaction data is eliminated by Cut-Through rather than retained indefinitely, and only the cryptographic proofs required for validation persist. Rising usage therefore amplifies the benefit — the more frequently value moves, the more intermediate state cancels — and long-term chain growth is governed far more by the size of the ledger's present state than by the volume of its payment history, with kernels as the sole per-transaction exception, addressed below. Both throughput levers — block weight and block time — are consensus policy rather than architectural constants, adjustable by network upgrade as demand requires. The design intent is a scaling path from today's millions per day toward billions per day; reaching that scale is an explicit research programme rather than a property of today's constants — parameter increases compounded with the kernel-compression work of Section XI, and the block-propagation, validation, and state-growth engineering to match — since each independently constructed transaction contributes one kernel under the current protocol. What the architecture fixes is the cost of throughput: each payment leaves approximately 106 bytes of permanent data (its kernel), several times less than a transparent-chain transaction and more than an order of magnitude less than ring-signature or shielded-proof designs, while spent outputs are pruned entirely. Throughput therefore does affect storage today — kernels accumulate with use, as they must for verifiability — but Epic buys each unit of throughput with materially less permanent state than competing architectures, and kernel accumulation itself is an active research target rather than an accepted endpoint (Section XI). The design intent remains scaling everyday payments at the base layer rather than delegating them to custodial or secondary layers.

VIII · Monetary Policy and Emission

Epic's emission is defined in consensus code and follows a fixed schedule of eras:

ERA	BLOCK HEIGHTS	APPROX. DATES	BLOCK REWARD	CUMULATIVE SUPPLY	% OF 21M
1	1 – 480,960	Sep 2019 – Aug 2020	16 EPIC	7,695,360	36.6%
2	480,961 – 1,157,760	Aug 2020 – Nov 2021	8 EPIC	13,109,760	62.4%
3	1,157,761 – 2,023,200	Nov 2021 – Jul 2023	4 EPIC	16,571,520	78.9%
4	2,023,201 – 3,175,200	Jul 2023 – Sep 2025	2 EPIC	18,875,520	89.9%
5	3,175,201 – 4,642,560	Sep 2025 – Jul 2028	1 EPIC	20,342,880	96.9%
6+	4,642,561 –	from 2028	0.15625 EPIC, halving every 2,102,400 blocks	→ 21,000,000	100%

As of July 2026 the network is in Era 5, emitting 1 EPIC per 60-second block. The schedule is deterministic under the current consensus rules — enforced and verifiable by every node — and alterable only by a consensus-breaking upgrade voluntarily adopted across the network; no authority exists that could impose a change. In May 2028 Epic's cumulative supply draws level with Bitcoin's — the Singularity — after which both currencies follow effectively identical asymptotic paths toward their nominal 21-million maximums. Each Epic divides into 100,000,000 freeman. The result is an issuance profile closely aligned with Bitcoin's, achieved without premine: Bitcoin's first nineteen years of emission are compressed into Epic's first nine, placing the great majority of supply into open, mineable circulation while the network is young.

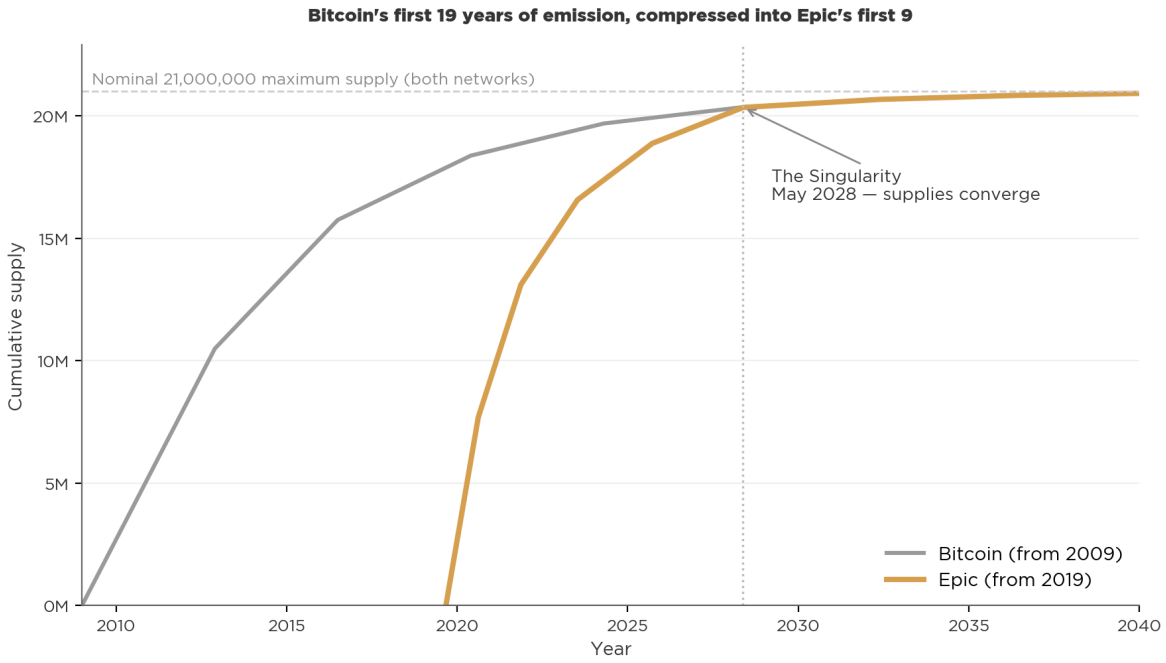

Figure: Bitcoin's first 19 years of emission, compressed into Epic's first 9. The two supplies converge at the Singularity (May 2028) and follow effectively identical paths toward the nominal 21,000,000 maximum. Source: consensus emission schedules of both networks — Epic per the era table above (consensus.rs); Bitcoin per its 210,000-block halving schedule.

The long-term security budget. A proof-of-work network's security is bounded by its security budget [9] — the total value it pays miners, which caps what rational miners can spend and therefore what an attacker must outspend. Bitcoin's security budget declines by design: the block subsidy follows a negative exponential, and transaction fees have not grown to replace it as activity migrates to off-chain and custodial layers. Critically, **whatever budget remains must fund the manufacture and perpetual replacement of single-purpose hardware that has no reason to exist otherwise.**

The energy miners consume is not the dependency — electricity is simply the medium through which miners pay for the coins they earn. The dependency is the machinery and the industrial plant around it: industry analyses sized the global Bitcoin ASIC hardware market at roughly $11 billion for 2025 [11] — a proxy for the scale of dedicated-hardware spend, with individual public miners' filings disclosing unit purchases at hundred-million-dollar scale — plus the dedicated facilities built to house, power, and cool a fleet that can do nothing but compute one hash function. That capital produces nothing else, depreciates to obsolescence in three to five years, and must be re-bought continuously — funded wholly by block subsidy and fees, the very revenue stream that halves every four years. Bitcoin's security rests on the assumption that a perpetually shrinking income will keep justifying a perpetual re-investment in perishable, single-purpose plant.

Epic carries the same scarcity profile with a structurally different security foundation. Its hashpower is drawn from hardware whose acquisition, energy infrastructure, and replacement cycles are already economically justified by non-mining uses, so the security budget needs to cover only the marginal cost of participation — not the existence of the fleet itself. The claim, stated as the tradeoff it is: Epic does not guarantee a security floor — no proof-of-work network can, since realized hashrate always depends on what the reward attracts. Commodity hardware lowers entry

costs, lets many participants mine with equipment they already own, and — because the devices retain resale value and non-mining utility — reduces stranded-capital risk relative to single-purpose ASICs; realized security nevertheless remains a function of active hashrate, energy and opportunity costs, miner concentration, and the availability of rentable or compromised compute. An ASIC network's sunk capital cuts the other way, making committed hashrate observable and costly to redirect; the two models trade different risks. Epic's position is comparative, not absolute: less of its security budget must be consumed re-buying single-purpose plant, the potential supply of hashpower persists at scale regardless of coin price, and for a fixed-supply currency intended to function across a century of declining subsidy, that is the sounder structural bet than perpetually repurchasing one's own foundation.

IX · Protocol Evolution

Epic has executed three major network upgrades since launch — including consensus-level changes — each proposed in the open, implemented in open-source code, and adopted by the voluntary action of independent node operators and miners — with no coordinating company or foundation.

v3 (March 2022). A major re-architecture of the node: modernized codebase and dependencies, substantially faster and more reliable chain synchronization, improved peer discovery, and expanded APIs — the foundation on which subsequent upgrades were built.

v3.4 (June 2023). A consensus-level change to mining policy, setting the polyphasic allocation at 48% RandomX / 48% ProgPoW / 4% Cuckatoo31+ going forward, alongside a new header-sync process that further accelerated initial chain download. Subsequent 3.5–3.6 releases added blockchain checkpoints, fast-sync options, and node performance improvements.

v4 (November 2025; current release 4.0.3, June 2026). A substantial privacy-infrastructure upgrade: Tor process integration, onion-service support, and peer onion-address exchange via the new ONIONSTEM capability (Section V) — the groundwork for fully onion-routed transaction propagation — together with a comprehensive modernization of APIs, dependencies, and security posture.

The upgrade history demonstrates the governance model in practice: a network with no owner has nonetheless re-architected itself three times, changed consensus policy, and shipped a major privacy layer — because open-source coordination among voluntary participants is sufficient, exactly as Bitcoin has demonstrated for seventeen years.

X · Governance

Epic is governed the way Bitcoin is governed: by no one, which is to say by everyone who chooses to run its software. Stated precisely: no entity can unilaterally alter the network's consensus rules, and a change takes effect only when independently operated nodes and miners choose to adopt compatible software. Coordination, to be clear, exists — maintainers review code and cut releases, community operators run explorers, seed nodes, and relays, and documentation such as this paper is authored by community contributors. None of these roles carries authority over consensus: a maintainer's release binds no one who declines to run it. The accurate claim is no unilateral control over consensus, not no coordination. The corresponding process-level accountability, stated without identifying individual key-holders:

FUNCTION	HOW IT OPERATES	AUTHORITY OVER CONSENSUS
Code & releases	Public repositories under the EpicCash GitHub organization; changes land through maintainer review; releases are tagged from the organization	None — a release binds only those who choose to run it
Consensus upgrades	Proposed and discussed in public; shipped as versioned releases; activated only by adoption across independent miners and node operators	None unilaterally — adoption is the activation
Infrastructure (explorer, seeds, relays, domains)	Operated by independent community participants; every function has permissionless alternatives — any node, any relay, self-hosted tooling	None — not consensus-critical
Fair-launch contribution	Recipient outputs protocol-committed in `foundation.json`; per-block share fixed in consensus code; stream ends December 2027	None — protocol-defined and expiring
Wrapped EPIC (Solana)	Being developed by an independent ecosystem participant; does not change or affect the Epic protocol in any way — optional additional utility for users who choose it; disclosures to accompany launch	None — external to the protocol

Funds and custody, stated plainly. Epic has no protocol-controlled treasury and no incorporated foundation treasury — and, to be equally plain about provenance: the historical levy proceeds do not sit anywhere today. They were deployed contemporaneously, over the bootstrap years, on development, infrastructure, and integrations, and the funding allocation that produced them — 6.10% of issuance in total — terminates permanently at block height 4,377,600 in December 2027. What exists now is unrelated to the levy: small, initiative-specific funds raised independently by the community members leading particular efforts — an exchange integration here, website maintenance there — each self-funded for its own objective, outside consensus, and holding no authority over the protocol. These arrangements are informal by nature — no conflict-of-interest policy, no periodic audit — disclosed here as the trust assumption they are, and sized accordingly: minor sums, specific purposes, and no ongoing emission behind them. The levy's on-chain side is protocol-committed in `foundation.json`; its off-chain handling has been, likewise, a matter of community trust rather than formal control. There is no Epic company, no foundation, no protocol-controlled treasury, no on-chain voting apparatus, and no legal entity with authority over the protocol. Up-

grades are proposed in the open, implemented in open-source code, and adopted only when the network's independent node operators and miners choose to run them — a process that has successfully executed three major network upgrades without a central authority.

This was not the original plan, and the difference is worth stating plainly. The 2019 whitepaper anticipated a foundation and, later, a distributed autonomous structure. Between 2020 and 2021, the community's direct experience with formalized governance experiments demonstrated what the broader industry has since learned repeatedly: governance structures, treasuries, and voting systems are attack surfaces — social as much as technical. Foundations can be pressured, sued, captured, or corrupted; token voting concentrates power it claims to distribute. Epic responded by deliberately adopting the only governance model with a multi-decade record of capture resistance: Bitcoin's — making Epic one of the very few post-Bitcoin networks to actually replicate that model rather than merely invoke it. The absence of formal structure is not an omission. It is the design.

The fair-launch contribution. Every network must fund its own bootstrapping. The industry's dominant answers have been premines and venture allocations — instant, opaque, and permanent transfers of supply and control before the public ever participates. Epic chose a different instrument: a small, protocol-defined, time-limited share of block emission, fully specified in the consensus code and auditable by anyone. The contribution began at 8.88% of block rewards for the network's first 120 days, then declined on a fixed annual schedule — 7.77%, 6.66%, 5.55%, 4.44%, 3.33%, 2.22%, 1.11% — and stands at 1.11% today. Per the consensus rules, the levy ends at height 4,377,600 — late December 2027, months before the Singularity. Cumulatively it totals 1,280,478 EPIC — approximately 6.10% of terminal issuance — disclosed here alongside the per-block schedule. The comparison that number invites deserves precision: allocations elsewhere in the industry, Zcash's 20% Founders Reward being the canonical example, were structured as return vehicles for the investors who financed a company. Epic's contribution had no investors to repay. It was a bootstrapping instrument for early contributors building infrastructure in the absence of any company — and it expires rather than compounds. The difference is not merely six percent versus twenty; it is the difference between a funding tool that ends and a profit center that renews. Miners have received no less than 91% of emission at any point in the network's history, and currently receive approximately 98.9%.

In the absence of any incorporated entity, these funds were deployed informally by early contributors to support development, infrastructure, exchange integration, and ecosystem growth during the network's bootstrap years; the ongoing residual stream continues to support infrastructure and development. Compared against the industry's standard mechanisms, the structural properties speak for themselves: gradual rather than instant, protocol-encoded rather than discretionary, declining rather than permanent, and expiring — by consensus rule — on a schedule set before the first block was mined.

Institutional threat vectors: hostile accumulation. Capture resistance extends beyond the protocol to the asset itself. A defining trend of the mid-2020s is state accumulation of digital assets — most visibly the United States' establishment of a Strategic Bitcoin Reserve in 2025, capitalized not through purchases but with coins already in government custody through criminal and civil forfeiture. Civil asset forfeiture permits seizure without conviction, and on a transparent ledger the acquisition strategy is operationally straightforward: chain-surveillance contractors map holdings to identities, enforcement prioritizes the largest and most exposed positions, and seized units flow

into state reserves — every seizure simultaneously an acquisition. Epic's architecture interacts with this machinery by starving it of inputs. There is no rich list to triage, no address graph to subpoena, no transaction history from which to construct a forfeiture narrative, and no taint by which downstream holders can be implicated for the provenance of coins accepted in good faith. A state can still compel an individual or seize a device — no monetary design prevents that — but portfolio-scale, surveillance-driven accumulation requires surveillance, and Epic's ledger produces none. This is what unconditional money means in practice: an asset whose custody cannot be reassigned at scale through observation and administrative process. Individual compulsion remains — as acknowledged — but it does not compound into a reserve strategy.

XI · Development Roadmap

Active and planned work extends the protocol along the lines its architecture already establishes:

Wrapped EPIC on Solana (independent ecosystem initiative — in active development) — a wrapped representation of Epic on the Solana network, extending Epic's reach into high-throughput DeFi and exchange environments while the base layer remains private and unchanged. Wrapped instruments are ecosystem products built by participants; the Epic protocol exists independently of any wrapper, and the initiative's custody, backing, and redemption disclosures will accompany its launch.

Transaction reliability (in development) — continued hardening of the interactive transaction flow across all transports, reducing failure modes in wallet-to-wallet communication.

Parallel initial block download (PIBD) (in development within the Mimblewimble/Grin ecosystem, for adoption by Epic) — adoption of the Mimblewimble ecosystem's parallel state-sync mechanism, in which a new node downloads the chain state in independently verified segments from many peers simultaneously rather than as a single archive from one, making the already-light full-node bootstrap faster and more robust.

Nostr transaction relay (research) — migration of the Epicbox relay function to the Nostr protocol, replacing dedicated relay infrastructure with a decentralized network of independent, censorship-resistant relays carrying encrypted transaction slates.

Multisignature outputs (research, within the Mimblewimble/Grin ecosystem) — n-of-n multiparty control is native to Mimblewimble: partial Schnorr signatures simply add, so a jointly held output is an ordinary output requiring all parties to sign. Threshold control (t-of-n, e.g. 2-of-3) is a different problem — Mimblewimble has no script layer — and may be implemented with a secp256k1 threshold-signature protocol such as FROST (Flexible Round-Optimized Schnorr Threshold Signatures, RFC 9591) [29] — subject to an Epic-specific design covering distributed key generation, signer coordination, nonce handling, and recovery, plus implementation and audit. The output is a standard single Schnorr signature, so on chain, threshold custody would be indistinguishable from any other kernel. Highly available threshold receiver nodes of this kind are also the identified path for automated custodians and bridges, for whom interactive receipt otherwise imposes an always-on availability requirement that fire-and-forget networks do not.

Atomic swaps (research) — potential research targets include Monero, shielded Zcash, and Litecoin's MWEB, executed via adaptor signatures ("scriptless scripts", Section IV) [32], in which authorization is bound to the leakage of a secret witness from a counterparty's signature, enforcing the swap without contracts or scripts. Each pairing requires its own protocol construction — Monero's ed25519 signatures, Zcash's shielded pool, and MWEB differ in curves and schemes — no Epic pairing should currently be regarded as production-ready, and the design goal, to be demonstrated pair by pair, is that completed swaps appear on each chain as ordinary transactions. A privacy-to-privacy swap corridor has no dependence on centralized exchange infrastructure — the same infrastructure whose fragility Section III documents.

Kernel compression research — the kernel is the one per-payment artifact that accumulates permanently (~106 bytes). Documented research directions in the Mimblewimble ecosystem address it: non-interactive half-aggregation of kernel Schnorr signatures at block construction, summing the

signature scalars block-wide and cutting stored signature data substantially; horizon-based history compression, in which deeply buried kernel history is represented by its MMR root and aggregate excess under an assume-valid trust model, as new nodes already rely on header-committed roots during fast sync; and transaction aggregation ahead of block inclusion, reducing kernels per economic payment. None of these is deployed on Grin, MWEB, or — to public knowledge — any other Mimblewimble network. The boundary is quantifiable: at ~106 bytes per kernel, roughly 9.4 million transactions add one gigabyte of permanent data, so sustained volume in the region of ten million transactions per day (~1 GB/day) is the threshold by which compression must be deployed for the small-node guarantee to survive mass adoption. Epic accordingly treats kernel compression as critical-path engineering for the scaling roadmap, not optional research. Alongside kernel work, successor range-proof systems — Bulletproofs+ and Bulletproofs++, offering smaller proofs and faster batch verification while remaining setup-free — are natural upgrade candidates the research agenda tracks.

Post-quantum research — stated plainly: Epic is not quantum-resistant today. Its signatures and the binding of its commitments rest on the elliptic-curve discrete-logarithm assumption, so a cryptographically relevant quantum adversary would threaten signature unforgeability and, through commitment binding, supply integrity itself. One property genuinely survives such an adversary: Pedersen commitments are perfectly hiding, so the recorded ledger reveals no amounts even to unbounded computation — but hiding without binding is not money. Research therefore targets post-quantum commitment, proof, and signature constructions, the consensus upgrade to deploy them, and a migration path for existing outputs — all of which must be in place before a practical quantum adversary exists. On the proof-of-work side, community research is exploring a candidate fourth proof-of-work algorithm inspired by the lattice operations used in CRYSTALS-Dilithium — the mathematics standardized as ML-DSA in NIST FIPS 204 [35]. No public construction is currently available, so the work should be treated as exploratory rather than specified or ready for deployment. A post-quantum-mathematics mining member would diversify the hardware and cryptanalytic base of the polyphasic framework; it would not by itself make transactions quantum-resistant — the signature and commitment migration above remains the core of the research programme.

XII · Security Considerations

Fork choice and finality. Each header records Epic's cumulative difficulty as a per-algorithm vector. A block's proof is validated against the target for its scheduled algorithm — selection is deterministic, via the implementation's Feijoada policy engine — while chain ordering follows the implementation's aggregate ordering of the cumulative difficulty vector. Safety and liveness are therefore distinct properties: because the required algorithm for each slot is deterministic, loss or withholding of the scheduled algorithm's hashpower can delay block production even by an attacker who cannot construct the highest-work alternative chain. Reorganization resistance and liveness depend jointly on the schedule, per-algorithm hash availability, difficulty dynamics, and accumulated-work comparison; Epic accordingly does not have a single universal "51%" threshold, and this paper does not claim that every attack requires simultaneous majority control of all three algorithms.

Formally: at height h, the node derives the full per-algorithm target vector **d(h) = NextDifficulty(h, a(h−1), H(<h))** by copying the previous vector and retargeting exactly one component — the one belonging to the preceding block's algorithm a(h−1) — from 60 observations of that algorithm. The header then accumulates the **entire** vector, **T(h) = T(h−1) + d(h)**; the block's proof must satisfy **ProofDifficulty(h)[a(h)] ≥ d(h)[a(h)]** for its Feijoada-scheduled algorithm a(h); and competing chains — not necessarily at equal height — are ordered by the implementation's aggregate ordering of the cumulative vector (`Difficulty::Ord`, the component sum). An illustrative five-block instantiation in arbitrary units, starting from d = (RX 100, PP 90, CT 5) and T = (1000, 900, 50):

HEIGHT	SCHEDULED A(H)	RETARGETED COMPONENT (= A(H−1))	D(H) (RX, PP, CT)	T(H) CUMULATIVE	AGGREGATE
h+1	ProgPoW	RandomX → 101	(101, 90, 5)	(1101, 990, 55)	2,146
h+2	RandomX	ProgPoW → 89	(101, 89, 5)	(1202, 1079, 60)	2,341
h+3	ProgPoW	RandomX → 102	(102, 89, 5)	(1304, 1168, 65)	2,537
h+4	Cuckatoo	ProgPoW → 90	(102, 90, 5)	(1406, 1258, 70)	2,734
h+5	RandomX	Cuckatoo → 6	(102, 90, 6)	(1508, 1348, 76)	2,932

Non-retargeted components carry forward unchanged, every height adds the full vector, and each block's proof is checked only against its own scheduled component. Chain selection compares aggregate cumulative ordering — in principle, a shorter chain can win if its ordering is higher. Withholding a scheduled algorithm's hashpower can therefore degrade liveness without necessarily being sufficient to construct the highest-ordering alternative chain. The units above are illustrative; the mechanics mirror the v4.0.3 implementation. Finality is probabilistic, as in all proof-of-work systems: at 60-second blocks, confirmations accrue ten times faster in count than Bitcoin's, though equivalent settlement security is a function of accumulated work and attack cost, not block count alone. Separate safety and liveness analyses with simulations of single-algorithm withholding, sudden hashrate loss, and private-chain construction remain planned work.

Synchronization and trust model. A new node downloads headers first, then obtains the chain state — unspent outputs with their range proofs, and the kernel set — and verifies it against roots committed in every header. Header-chain assurance is mode-dependent, and this paper states the shipped behavior exactly. v4.0.3 still checks hard-coded checkpoint hashes when encountered; with the default `skip_pow_validation = true` and `disable_checkpoints = true`, however, header-sync proof-of-work and difficulty-increment validation are skipped beyond as well as within the checkpointed range, and the full-block synchronization path re-verifies proof-of-work only for a recent window. The default therefore optimizes bootstrap speed at a materially weaker assurance during initial sync, raising exposure to eclipse-style bootstrap attacks. Setting `skip_pow_validation = false` enables header proof-of-work and difficulty validation from genesis — full header-PoW validation, which is not a replay of transaction bodies already discarded by pruned state synchronization. For institutional operators, bridge validators, oracles, and cross-chain infrastructure, full header-PoW validation should be treated as a baseline security requirement — together with peer diversity and out-of-band tip verification. The checkpoint list's last entry sits at height 2,200,000, well below the current tip; refreshing it, publishing its authorship and update policy, and revisiting the shipped defaults are identified engineering items. Epic maintains three append-only Merkle Mountain Ranges — one each for outputs, range proofs, and kernels — and MMRs yield compact inclusion proofs against those header-committed roots. This is what lets a pruned node trust current state without replaying history from genesis, lets Cut-Through discard data without weakening the commitment structure, and is precisely the structure PIBD (Section XI) parallelizes. A newly synchronized pruned node verifies — in full header-PoW-validation mode — accumulated header work from genesis; in all modes it verifies current-state and MMR-root consistency, all retained (unspent) output range proofs, all retained kernels and their signatures, and the global balance equation: the sum of unspent commitments minus authorized emission must equal the sum of kernel excesses (plus the aggregate offset). What it does not do is independently re-execute validation against historical objects that have already been pruned: the roots commit to that history, while the validity of discarded data rests on the accumulated-work chain and on validation performed by the network when the data was available. This assurance model differs from Bitcoin's pruning, which downloads and validates all historical blocks before discarding old bodies; Epic's state-sync trades that replay for root-committed state at a fraction of the cost. Nodes run pruned by default, retaining only what verification requires; an optional archive mode retains full history. The planned parallel initial block download (Section XI) distributes state sync across many peers.

Spam and denial-of-service. Block weight is capped at 40,000, and the reference node's default relay policy uses an `accept_fee_base` of 0.001 EPIC per weight unit — configurable local transaction-pool policy, not a consensus-enforced minimum — so a typical weight-47 transaction faces a default relay threshold of 0.047 EPIC, committed in the kernel and paid to the including miner. Relay fees price third-party spam at the mempool boundary; a miner can include its own zero-fee transactions directly, bearing opportunity cost rather than fees — so the floor governs relay admission, not consensus validity or miner-authored bloat, and kernel compression (Section XI) is future mitigation rather than a current control. One asymmetry deserves naming: kernels are the artifact spam permanently leaves behind (~106 bytes each), so cheap relay makes long-run ledger growth an attack the attacker pays for once and every archival node pays for indefinitely. Operators can raise `accept_fee_base` locally — and this asymmetry is precisely why the kernel-compression direc-

tions of Section XI, half-aggregation and horizon-based history compression, are security work rather than mere storage optimization. Peer-level defenses include capability handshakes, peer diversity in connection selection, and standard ban scoring for misbehaving peers.

Mining-related risks. Pool concentration within a single algorithm is the perennial proof-of-work watch item; Epic's mitigation is structural, since even total capture of one algorithm yields a minority of block production — with the liveness caveat stated above under fork choice. The flip side of CPU-friendly mining — botnet participation — is acknowledged and bounded the same way: RandomX earns at most its 48% share, under its own difficulty. Per-algorithm retargeting — 60 observations of each algorithm — absorbs hashpower shocks over roughly 125 minutes for the 48% algorithms and roughly 25 hours for Cuckatoo31+, far faster than fixed-epoch schemes, subject to the liveness caveat under fork choice above.

Wallet and relay failure modes. Interactive transactions can stall when a counterparty goes offline mid-construction; Epicbox's asynchronous relay removes the simultaneity requirement, unfinished transactions can be cancelled without loss, and transaction-reliability hardening is an active roadmap item. Relay unavailability degrades to alternative transports — direct Tor, file exchange — rather than to failure, and the planned Nostr migration reduces reliance on dedicated Epicbox relay infrastructure by allowing transactions to use a broader decentralized relay ecosystem.

Independent audit history. The component lineage Epic is built upon has undergone eight separate independent third-party security audits [13–20]. RandomX was audited four times in 2019 — Trail of Bits (May), Kudelski Security (July), X41 D-SEC (July), and Quarkslab (August). ProgPoW was audited by Least Authority (September 2019). The Grin codebase from which Epic derives received a secp256k1-zkp extensions review by Jean-Philippe Aumasson (December 2018), a Coinspect security audit (October 2019), and an independent review by Nym Seddon (August 2020). On version provenance: the RandomX and ProgPoW audits covered the reference algorithm implementations, which Epic consumes through maintained Rust bindings (EpicCash/randomx-rust, EpicCash/progpow-rust); the Grin reviews covered the 2018–2020 codebase from which Epic forked, and the core Mimblewimble validation logic and secp256k1-zkp cryptographic library remain substantially that audited lineage. Epic-specific consensus additions — polyphasic scheduling, the emission schedule, and the levy mechanism — postdate those audits and fall outside their scope. These reviews do not substitute for an audit of Epic as an integrated system — Feijoada scheduling, polyphasic interactions, emission logic, and state synchronization together — which has not yet been performed. Commissioning such an end-to-end review is a declared maturation milestone on the community's path, not an open question of whether but of scoping and funding.

Liveness monitoring and cross-chain buffers. Because algorithm scheduling is deterministic, integrators — bridges, oracles, and wrapped-asset systems — should treat per-algorithm block-production gaps as a monitored signal and size their confirmation buffers to weather temporary stalls; a pure withholding episode delays the chain; confirmation buffers reduce downstream exposure but do not eliminate reorganization or bridge-specific risk.

XIII · Protocol Parameter Summary

The following consolidates the protocol's operative constants as of v4.0.3 (values verifiable in `core/src/consensus.rs` of the EpicCash/epic repository).

PARAMETER	VALUE
Consensus protocol	Mimblewimble (Grin-derived implementation, Rust)
Elliptic curve	secp256k1 (Pedersen commitments; Schnorr signatures)
Range proofs	Bulletproofs, no trusted setup (675 bytes per output)
Block target	60 seconds (1,440 blocks/day)
Proof-of-work	RandomX 48% / ProgPoW 48% / Cuckatoo31+ 4% (policy since v3.4)
Difficulty adjustment	Per algorithm; recalculated every block over a rolling 60-block window of that algorithm (~2 h wall time at 48% share)
Maximum block weight	40,000 (input = 1, output = 21, kernel = 3) ffi 851 typical transactions per block
Theoretical capacity	ffi 1.2 million transactions/day at current parameters
Coinbase maturity	1,440 blocks (~24 hours)
Cut-Through horizon	10,080 blocks (~7 days)
Permanent data per transaction	~106 bytes (kernel); spent outputs pruned
Unspent output size	~709 bytes (commitment + 675-byte range proof + features)
Nominal maximum supply	21,000,000 EPIC
Smallest unit	1 freeman = 0.00000001 EPIC
Emission	Fixed era schedule (Section VIII); deterministic under current consensus rules
Current node release	v4.0.3 (June 2026)

XIV · References

1. S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
2. T. E. Jedusor, Mimblewimble, 2016.
3. A. Poelstra, Mimblewimble (extended treatment), 2016.
4. B. Bünz et al., Bulletproofs: Short Proofs for Confidential Transactions and More, IEEE S&P, 2018.
5. G. Fanti et al., Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees, ACM SIGMETRICS, 2018.
6. tevador et al., RandomX: Proof-of-work algorithm optimized for general-purpose CPUs, specification, 2019.
7. IfDefElse, ProgPoW: A Programmatic Proof-of-Work, specification, 2018.
8. J. Tromp, Cuckoo Cycle: A Memory Bound Graph-Theoretic Proof-of-Work, 2015.
9. J. McKinney, Bitcoin Security: a Negative Exponential, 2018.
10. CVE-2010-5139, Bitcoin value overflow vulnerability, 2010.
11. Cognitive Market Research, ASIC Bitcoin Mining Hardware Market, 2025; CoinLaw, Cryptocurrency Mining Hardware Market Statistics, 2024; hardware lifetimes per public mining-company depreciation schedules (3–5 years).
12. U.S. executive order establishing the Strategic Bitcoin Reserve, March 2025.
13. Trail of Bits, RandomX Security Assessment, May 2019.
14. Kudelski Security, RandomX Security Audit Report, July 2019.
15. X41 D-SEC, RandomX Security Audit, July 2019.
16. Quarkslab, Security Audit of Monero RandomX, August 2019.
17. Least Authority, ProgPoW Algorithm Final Audit Report, September 2019.
18. J.-P. Aumasson, secp256k1 Extensions Security Review (Grin), December 2018.
19. Coinspect, Grin Security Audit, October 2019.
20. N. Seddon, Grin Independent Security Review, August 2020.
21. Epic Cash source repositories and release history, github.com/EpicCash, 2019–2026.
22. Epic Cash network explorer, explorer.epiccash.com, 2026.
23. M. Freeman, community incident account and reimbursement claim form, Epic Cash on Medium, April 2021 — primary account of the March 2021 value-overflow incident (window March 12–20, 2021; ~2.8M counterfeit EPIC reaching ViteX; ~6.6 BTC in trader losses; ~200,000 EPIC in miner revenue losses). medium.com/epic-cash/epicenter-iou-and-vitex-claim-form-59432315668b
24. Epic node releases v2.13–v2.15 (March–April 2021), github.com/EpicCash/epic/releases — remediation-era releases.
25. Royal Canadian Mounted Police, press release on the seizure and dismantling of the TradeOgre exchange, September 18, 2025 (primary); CoinDesk contemporaneous coverage (secondary).
26. Service cessation of BiHODL, CITEX, VinDAX, and ViteX: Internet Archive captures of the respective exchange domains.
27. Bitcoin Core documentation — pruned node disk requirements (prune mode, ~7 GB).
28. Historical Bitcoin transaction-fee trackers; reported peak values vary with measurement methodology (daily vs. intraday windows, mean vs. median).
29. IETF RFC 9591, The Flexible Round-Optimized Schnorr Threshold (FROST) Protocol, 2024.
30. H. Chung et al., Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger, 2020.
31. L. Eagen, Bulletproofs++, 2022.
32. A. Poelstra, Scriptless Scripts (adaptor signatures), 2017.

33. Grin project documentation, Parallel Initial Block Download (PIBD).
34. Nostr protocol, NIP-01 and relay specifications.
35. NIST FIPS 204, Module-Lattice-Based Digital Signature Standard (ML-DSA), derived from CRYSTALS-Dilithium, 2024.
36. Aggregate installed-base estimates for consumer CPUs and GPUs across PCs, workstations, and smartphones (industry shipment analyses).