CYPHERMINT

Mainnet V3 Whitepaper

Technical and Economic Description

“A blockchain is built by those who stay to build and verify it.”

Version 1.1
July 2026
Asset symbol: CPM

MAINNET V3 - NEW SOFTWARE - PARTICIPATE AT YOUR OWN RISK

Document status

Purpose. This document describes CypherMint Mainnet V3, its consensus identity, ASERT difficulty adjustment, tokenomics, operating model, security assumptions and development direction.

Release identity. Mainnet V3 is identified by network ID cyphermint-mainnet-v3 and genesis hash 000000078daa263b7c52fb55ab87dd4c9e2e97c2baad46adad96cff322634e73.

Risk. CypherMint is new software. No financial return, exchange support, liquidity, legal status or uninterrupted operation is promised. Wallet owners are responsible for custody and backups.

Contents

1. Abstract and motivation

CypherMint is a native UTXO-based proof-of-work blockchain designed around transparent validation, open participation and operational simplicity. Mainnet V3 combines a full validating node, native mining-pool service, open-source miner, encrypted wallets, signed transactions, peer synchronization, chainwork-based fork selection, public APIs and a block explorer.

The project uses double-SHA-256 proof of work, a ten-minute target interval, per-block ASERT difficulty adjustment, halving-based issuance, coinbase maturity and greatest-valid-chainwork selection. The permanent reward allocation is 97% residual to miners, 1% to the pool or node operator and 2% to the development address.

Mainnet V3 begins from a fresh zero-output genesis block. Normal issuance begins with block 1. Independent nodes validate the same published network ID and genesis hash. The open native mining protocol is intended to give home miners an early opportunity, but CypherMint does not claim permanent ASIC resistance.

Design principles

- Validation before convenience: the chain and consensus rules are authoritative; caches and indexes are disposable.
- Open participation: anyone can create a wallet, run a node, mine and verify public chain data.
- Explicit economics: reward allocation and development funding are visible consensus rules.
- Clean network identity: releases publish genesis, network identity and source/release hashes.
- Incremental development: wallets, exchanges and games follow a stable core rather than replace it.

2. Scope of Mainnet V3

Mainnet V3 is a public CypherMint network with a zero-supply genesis and ASERT difficulty adjustment, intended for open participation and independent verification:

- Can independent users install the software and create secure encrypted wallets?
- Can nodes discover peers, synchronize and recover after interruption?
- Can miners connect to local and remote pools and receive correctly allocated rewards?
- Can transactions be created, signed, relayed, confirmed and reorganized correctly?
- Does the public explorer accurately represent node state?
- Can public nodes remain usable through peer loss, restart and bootstrap-node outages?
- Are release identity, logs and diagnostics sufficient for independent operators?

Mainnet V3 remains new software. Implementation defects, custody errors, network concentration, incompatible upgrades and service outages can cause loss or disruption. No economic value or exchange listing is guaranteed.

3. System model and architecture

CypherMint uses an account-independent UTXO model. Transactions consume existing unspent outputs and create new outputs assigned to addresses. A wallet proves spending authority by signing a deterministic transaction payload with the corresponding private key.

Major components

Component	Role
Full node	Validates genesis, blocks, proof of work, difficulty, timestamps, transactions, reward allocation and chainwork.
Pool service	Exposes the mining protocol and HTTP API on a common service port.
Miner	Requests work, performs double-SHA-256 proof of work and submits shares or block solutions.
Wallet tools	Create, encrypt, inspect, back up, restore and use wallets to sign transactions.
Peer manager	Discovers, verifies, persists and replaces peers while enforcing network identity.
Derived indexes	Accelerate UTXO and chain lookups but remain rebuildable from the validated chain.
Explorer	Presents public blocks, transactions, addresses and network status without wallet secrets.

Trust order at startup

1. Load the expected genesis identity.
2. Load the saved active blockchain, if present.
3. Fully validate the saved chain.
4. Activate the validated chain.
5. Restore exact-tip snapshots or rebuild indexes from the chain.
6. Independently verify derived state.
7. Rebuild runtime caches.
8. Start API, pool and peer synchronization services.

Corruption refusal. If a live chain file exists but fails validation, the node refuses to silently restart from genesis. This prevents an operator from unknowingly discarding state or accepting corrupted local data.

4. Consensus identity and genesis

A CypherMint network is identified by both its network identifier and genesis hash. Nodes reject peers and chain data belonging to another identity.

Published Mainnet V3 identity

- Release ID: Cyphermint-Mainnet-V3; software version: V3; consensus version: 3
- Network profile: mainnet-v3; network ID: cyphermint-mainnet-v3
- Genesis hash: 000000078daa263b7c52fb55ab87dd4c9e2e97c2baad46adad96cff322634e73
- Genesis UTC: 2026-07-23T14:23:40Z; spendable supply: 0 CPM

- Bitcoin anchor: block 959264, hash 0000000000000000000001f988cd9c530c35e75f5f53b29eab2e50178ca1eeb03
- Initial public nodes: 165.22.3.186:4333, 64.227.116.54:4333 and 165.227.4.15:4333

Genesis and anchor policy. Block 0 contains no spendable outputs, so every wallet begins Mainnet V3 at zero and normal issuance starts at height 1. Bitcoin block 959264 is included only as independently verifiable chronological not-before evidence; CypherMint does not inherit Bitcoin security or consensus.

5. Proof of work, difficulty and timestamps

CypherMint uses double SHA-256 proof of work. A block is valid only when its independently calculated block-hash integer is less than or equal to the target encoded by its compact difficulty field.

- Canonical target conversion
- Maximum permissible target
- Chain-derived expected difficulty
- Proof-of-work verification on every block
- Independent block-work and cumulative-chainwork calculation

Parameter	Value
Target block interval	600 seconds
Difficulty algorithm	ASERT (aserti3-2d)
Adjustment frequency	Every block from height 2
ASERT half-life	172,800 seconds
Anchor / first adjusted block	Block 1 / block 2
Maximum/easiest target	0x1d180000
Median Time Past / future limit	11 blocks / 7,200 seconds

ASERT operation. Genesis and block 1 use the fixed starting target. Block 1 is the anchor and block 2 is the first ASERT-adjusted block. The aserti3-2d calculation runs after every block with a 172,800-second half-life and a 600-second target interval. Its reference schedule begins one target interval before block 1, so waiting between genesis creation and first mining does not bias difficulty. The target cannot become easier than 0x1d180000. Individual block times remain random; ASERT aims for a long-run ten-minute average.

Mining accessibility. CypherMint uses SHA-256D and does not claim permanent ASIC resistance. Its native mining protocol, lack of a standard Stratum endpoint and lack of an official ASIC adapter create temporary compatibility friction. The miner and protocol remain open source, so motivated operators may eventually build specialized-hardware compatibility. ASERT stabilizes block timing as hashrate changes; it does not exclude miners.

6. Chainwork, transactions and UTXOs

Chain selection

```
block_work = floor(2^256 / (target + 1))
```

Cumulative chainwork equals parent chainwork plus the current block work. The active chain is the fully valid branch with strictly greater accumulated chainwork. Equal-work and lower-work branches do not replace the active chain.

Before committing a reorganization, a node identifies the common ancestor, validates the complete candidate branch, verifies its chainwork and only then replaces the active branch. Disconnected regular transactions may return to the mempool if still valid; disconnected coinbase transactions never do.

Transaction validity

- Every referenced output exists and is unspent.
- No input is duplicated or double-spent.
- The public key owns the referenced address.
- Every input signature is valid for the deterministic signing payload.
- Outputs use valid addresses and integer mint amounts.
- Output value does not exceed input value.
- Transaction, script and block limits are respected.

Transaction fees equal input value minus output value. Fees are included in the block reward and distributed by the same permanent allocation rule as the subsidy. The mempool rejects conflicting spends, treats duplicate submission idempotently, removes confirmed transactions and restores still-valid transactions after a reorganization.

7. Monetary policy and tokenomics

```
1 CPM = 100,000,000 mints
1 mint = 0.00000001 CPM
```

All consensus monetary calculations use integer mints rather than floating-point values.

Parameter	Value
Initial block subsidy	50 CPM
Halving interval	210,000 blocks
Genesis spendable supply	0 CPM
Normal issuance begins	Block 1
Nominal maximum subsidy issuance	20,999,999.9769 CPM

The exact maximum is slightly below 21 million CPM because integer-mint halving eventually reaches zero. Transaction fees redistribute existing CPM and do not create new supply.

Era	Reward	Blocks	Era issuance	Cumulative
0	50 CPM	210,000	10,500,000 CPM	10,500,000 CPM

Era	Reward	Blocks	Era issuance	Cumulative
1	25 CPM	210,000	5,250,000 CPM	15,750,000 CPM
2	12.5 CPM	210,000	2,625,000 CPM	18,375,000 CPM
3	6.25 CPM	210,000	1,312,500 CPM	19,687,500 CPM
4	3.125 CPM	210,000	656,250 CPM	20,343,750 CPM

Permanent reward allocation

Recipient	Nominal share	Purpose
Miner	97% residual	Proof-of-work security and block production
Pool/operator	1%	Public node, pool and infrastructure operations
Development address	2%	Core, wallet, explorer, testing and ecosystem development

50 CPM example
Miner: 48.5 CPM
Operator: 0.5 CPM
Development: 1.0 CPM
Total: 50.0 CPM

The miner receives the integer residual after the fixed portions, so any rounding remainder stays with the miner. The allocation is active from block 1 and remains a permanent protocol fixture. The zero-output genesis means the development wallet began with no pre-mined balance and receives its 2% share at address CUAoJEknQibsse3wZKT28vHbnkuEungQWQ only as blocks are mined.

Node operator rewards and wallet separation

A node's networking identity and its payout destination are separate. Each machine generates its own node identity, while --pool-address supplies only the public address that receives the operator allocation. The node does not need the private key, encrypted wallet or wallet password corresponding to that address.

Recommended custody model:

- Create and encrypt operator wallets on a trusted workstation or offline system.
- Place only public payout addresses on Internet-facing nodes.
- Keep passwords, wallet files and recovery material away from droplets.
- Use one address across several nodes for simple custody, or unique addresses for node-level accounting.
- Use separate operator and miner addresses when the same party both hosts a node and mines.

Operating a node does not create an automatic network-wide income stream. The 1% operator allocation applies to blocks constructed through that node or pool. If no miner produces a block through the pool, that node earns no operator allocation for the period.

A public pool can temporarily control miner escrow when miner amounts are below the direct-output threshold, exceed coinbase output capacity or include proportional rounding remainder. That value remains part of the miner allocation, must be tracked in the pool payout ledger and must not be treated as extra operator income.

Beyond compensation, public nodes provide independent validation, peer availability, chain relay, transaction propagation, public-query capacity and resistance to dependence on a single bootstrap service.

Coinbase maturity

Newly mined outputs cannot be spent immediately. Mainnet V3 coinbase maturity is 100 confirmations: spending is rejected before the maturity boundary and allowed after the required confirmations are reached.

8. Wallet custody and development funding

A wallet contains the private signing authority for an address. The secure wallet format supports password-based encryption and verifies that decrypted key material derives the stored address.

- Public addresses can be shared and queried anywhere.
- Private keys or encrypted wallets plus passwords control spending.
- Nodes and explorers do not need wallet files.
- Mining requires only a payout address.
- Backups must be verified and stored separately from passwords.
- Public balance lookup never reveals or recreates private keys.

The permanent development wallet uses an offline ceremony with a primary encrypted wallet, two portable encrypted backups, an independent restore proof and a secret-free public record.

Funding transparency

- Publish the development address.
- Keep receipts and spending publicly visible on-chain.
- Document custody and backup policy.
- Publish periodic development reports.
- Separate protocol-development funds from pool/operator income.
- Disclose material changes to spending policy.

9. Networking, APIs and explorer

Nodes maintain a bounded verified peer set. Saved verified peers are preferred before bootstrap fallbacks. Initial public Mainnet V3 nodes include 165.22.3.186:4333, 64.227.116.54:4333 and 165.227.4.15:4333.

Current public node	Service
165.22.3.186	TCP 4333
64.227.116.54	TCP 4333
165.227.4.15	TCP 4333

Bootstrap nodes are not consensus authorities. They only help a new node find peers. Every peer must match network ID cyphermint-mainnet-v3 and genesis hash 000000078daa263b7c52fb55ab87dd4c9e2e97c2baad46adad96cff322634e73. Peer exchange remains bounded and excludes non-public address classes from automatically shared peer lists.

Public API and explorer

The full node exposes read APIs used by command-line tools and the explorer. Public queries include node identity, network identity, chain tip, peers, mempool, balances, UTXOs and transaction lifecycle information.

- Display blocks and proof-of-work metadata.
- Display transactions and confirmations.
- Display public address balances and UTXOs.
- Expose network status and chain identity.
- Allow direct comparison with node results.

The explorer is not a wallet and must not hold private keys. The current public explorer is available at http://165.22.3.186:8080; use HTTP until a domain and TLS certificate are configured.

10. Security model and launch process

Derived state and recovery

UTXO and chain indexes accelerate reads. Persistent snapshots accelerate restart. Neither replaces chain validation. A snapshot is accepted only when integrity, network identity, genesis identity, consensus version, chain length, tip height and tip hash match the active chain. Missing, stale, corrupt or foreign snapshots are discarded and rebuilt.

Tested security properties

- Network ID and genesis-hash separation
- Full startup chain validation
- Double-spend prevention
- Signature and address-ownership checks
- Reward-split, proof-of-work and ASERT expected-difficulty enforcement
- Timestamp and Median Time Past rules
- Chainwork validation and strongest-chain selection
- Mempool conflict handling and reorganization restoration
- Coinbase maturity enforcement
- API rate limiting and request-size limits
- Bounded peer and relay workers
- Duplicate node-identity rejection
- Rotating operational logs
- Source and archive hashing

Testing is evidence, not a guarantee. Mainnet V3 remains subject to implementation errors, operational assumptions, custody failures, hashrate concentration and real-world network conditions.

Mainnet V3 release and launch record

1. Selected the proven working-system source baseline.

2. Assembled a clean release without wallets or live runtime state.

3. Retained the permanent development reward destination and zero starting balance.

4. **Froze the Mainnet V3 network identity, permanent economics and ASERT consensus parameters.**
5. **Published the initial bootstrap-node list.**
6. **Selected Bitcoin block 959264 as public not-before evidence and set an explicit UTC genesis time.**
7. **Generated and mined the deterministic zero-output Mainnet V3 genesis block.**
8. **Packaged the release as Cyphermint-Mainnet-V3.**
9. **Started clean node runtimes with unique identities.**
10. **Deployed public nodes and the browser explorer.**
11. **Tested block production, ASERT target changes and multi-node synchronization before resetting the release chain to genesis for public launch.**
12. **Continued synchronization, peer and restart observation across Windows, Linux/Kali and droplets.**
13. **Updated operator documentation and Quick Start guidance.**
14. **Opened participation to independent wallet users, miners and node operators.**

11. Development roadmap

Wallets

- Desktop graphical wallet
- Simplified secure backup and restore
- Hardware-wallet or external-signer support
- Watch-only addresses
- Transaction history and confirmation views
- Fee estimation
- Multi-address organization
- Mobile wallet research

Node and protocol

- Continued consensus test expansion
- Independent implementation review
- Deterministic serialization test vectors
- Stronger network-time handling
- Improved initial block download
- Service installers and packaging
- Monitoring dashboards
- IPv6 and mature DNS seeds
- Protocol compatibility policy

Explorer and developer platform

- Richer address and UTXO pages
- Transaction graphs and fee statistics

- Difficulty and hashrate estimates
- Documented read-only API
- SDK examples for Python, JavaScript and game engines
- Event subscriptions after security review
- Public status and incident reporting

Games and applications

Possible future uses include tournament rewards, player-to-player transfers, optional marketplace settlement, verifiable receipts and community mining events. Game integration is a direction, not a guarantee. Private keys must not be embedded in game clients, and integrations must separate game accounts from blockchain custody.

Exchanges and market infrastructure

Exchange readiness requires stable releases, deposit management, confirmation and reorganization guidance, reliable APIs, deterministic fee and balance accounting, monitoring, incident response, compliance review and independent security assessment. No listing is promised.

12. Governance, risks and conclusion

Protocol change

CypherMint currently has no automatic on-chain governance system. Protocol changes are implemented in software and require review, testing, publication and participant adoption. Changes to block or transaction validity, hashing, signatures, ownership, subsidy, reward allocation, difficulty, timestamps, chainwork, maturity or consensus limits require explicit consensus review and likely a consensus-version change or new network.

Risks and limitations

- Implementation defects can invalidate blocks or transactions.
- Python performance may limit throughput and mining efficiency.
- A small early network is vulnerable to concentration and outages.
- Bootstrap nodes can fail or become unreachable.
- Users can lose passwords or wallet files.
- Software upgrades can be incompatible.
- Incompatible software or consensus changes can split the network.
- Explorer or API bugs can display incorrect information.
- Economic value, exchange support and liquidity are not guaranteed.
- Laws may affect mining, custody, exchange access and application integration.

Conclusion

CypherMint does not promise a finished financial ecosystem. Mainnet V3 provides a transparent public chain from a clean, zero-supply genesis using a validating node, explicit economics, secure wallet custody, ASERT difficulty adjustment and a published identity.

The network depends on participants who run nodes, mine, verify, test, document and build useful software around it. Mainnet V3 keeps the implementation open source and makes that work publicly reproducible.

Appendix A - Consensus parameter summary

Parameter	Mainnet V3 value
Currency symbol	CPM
Base unit	100,000,000 mints per CPM
Proof of work	Double SHA-256
Target block interval	600 seconds
Difficulty algorithm	ASERT (aserti3-2d), 172,800-second half-life; block 1 anchor; block 2 activation
Initial subsidy	50 CPM
Halving interval	210,000 blocks
Nominal maximum subsidy issuance	20,999,999.9769 CPM
Miner allocation	97% residual
Pool/operator allocation	1%
Development allocation	2%
Fee treatment	Same permanent allocation
Genesis spendable output	0 CPM
Normal issuance	Begins at block 1
Chain selection	Greatest valid accumulated chainwork
Median Time Past	11 blocks
Maximum future timestamp	7,200 seconds
Coinbase maturity	100 confirmations
Default node/pool/API port	4333
Default explorer port	8080

Appendix B - Release-package boundary

Publish source and documentation, not live identity or custody files.

Include	Exclude
source/cyphermint.py and cyphermint_core	Wallet JSON files and passwords
runtime/genesis_block.json and static Mainnet V3 records	live_blockchain.json and node_identity.json
source/bootstrap_peers.json and requirements.txt	Snapshots, pool_peers.json and payout ledgers
Public explorer source when distributed separately	RPC tokens, logs and backups

Include	Exclude
README, whitepaper and Quick Start	Development .bak, .pyc and __pycache__ files
README, consensus specification, whitepaper	Development .bak, .pyc and __pycache__ files

Appendix C - Implementation evidence

- CypherMint consensus specification draft.
- Interactive startup test reports.
- Peer discovery and failover test report.
- Logging and status command test report.
- Expanded exchange-readiness and three-node integration test report.
- Multi-node Mainnet V3 operation and ASERT testing on Windows, Linux/Kali and public droplets.

Maintenance requirement. Update this whitepaper when the published repository, bootstrap endpoints, explorer URL or protocol status changes.