Whitepaper

Noticoin protocol, monetary policy, wallet privacy, map notes and mining operations.

`Technical specification · 1.0.0`

Noticoin is a RandomX proof-of-work network with a capped supply and a map integrated into its wallet. This specification describes consensus, transaction privacy, the publication format for map pins, paid-note delivery, the operator registry, fund accounting and pool operation. The tables distinguish consensus parameters from application settings. The timing and emission charts are models with stated assumptions.

01 Consensus and network identity

02 Proof of work and block validation

03 Supply and block subsidy

04 Confirmations and spendability

05 Wallet and transaction privacy

06 Map record format and lifecycle

07 Operator registry and epochs

08 Map records and note release

09 Fees and the fund

10 Mining and pool accounting

11 Verification and launch criteria

12 Implementation and operational guides

01

Consensus and network identity

Noticoin derives from Monero v0.18.5.4. It targets 45-second blocks and verifies RandomX from block 1 with protocol version 16. Version 17 activates at height 160. Network IDs are checked during P2P handshakes. Each network has its own genesis hash, ports and address prefixes.

A clean node derives network identity from its configuration and genesis. The P2P handshake carried over Levin checks the 16-byte network ID. Address prefixes distinguish standard, integrated and subaddresses. Operators can compare the block at height zero with this manifest before using a node for payment verification. Regtest uses isolated data and must be explicitly enabled in payment services that accept it.

Network	P2P	RPC	ZMQ	Address prefixes
mainnet	19080	19081	19082	111 / 112 / 113
testnet	29080	29081	29082	114 / 115 / 116
stagenet	39080	39081	39082	117 / 118 / 119

Standard / integrated / subaddress

Mainnet

Genesis hash

`2784467d7c909874668b9f0accf533c63f77aad79a3b72bfc5b23ca09e3cb614`

P2P network ID

`305639f9810544e48197a8f69a0a6d19`

Testnet

Genesis hash

`e2c8d4803df588a35e8289d2f939f3521ec92c43d4e8e453d05585917c205a90`

P2P network ID

`3946c278a80142618044dfa92619e1e7`

Stagenet

Genesis hash

`090c1b22ef96a0d7e05d7ec6321ad3f61d4f36c90cce2ce573e3c4be19f8daef`

P2P network ID

`0205f90fad5c4d41a8ad3bf27b9c1f06`

02

Proof of work and block validation

Nodes validate the parent, major version, timestamps, proof of work, transactions and the miner payout. They compare competing branches by cumulative difficulty. RandomX hashes the block hashing blob with a chain-derived seed. The first seed is the genesis hash. Default seed epochs are 2,048 blocks with a 64-block lag. These settings are distinct from the operator registry epochs.

Height	Major version	Scope
0	1	Genesis
1	16	RandomX · CLSAG · Bulletproofs+
160	17	6 Blocks

The adjustment takes the available history, limits it to 1,920 entries after the 40-block lag, sorts timestamps and trims the extremes when the sample is large enough. It takes the ceiling of cumulative work in the chosen interval times the target divided by elapsed seconds, with elapsed seconds floored at one. A history of at most one entry yields difficulty one. Separately, the timestamp rule uses a 160-block window and a two-hour future limit.

Parameter	Value
RandomX epoch / lag	2,048 / 64 Blocks
Difficulty window / lag / trim parameter	1,920 / 40 / 160 Blocks
Timestamp window / future limit	160 Blocks / 2 h

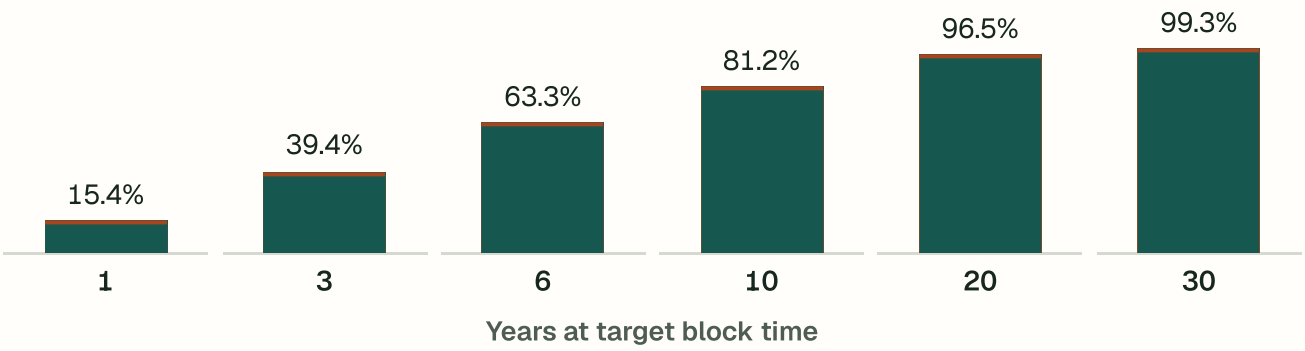
03

Supply and block subsidy

The cap is 382 million Noticoin at nine decimal places. Genesis counts 17.592186044 Noticoin in an unspendable output. After emission reaches the cap, only transaction fees remain as on-chain compensation for miners.

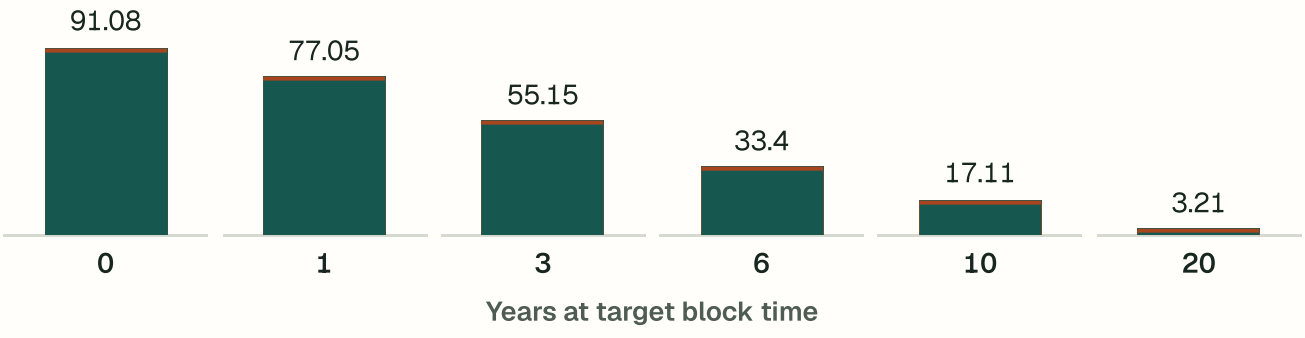
Let S be cumulative issued atomic units, including the genesis output. The unpenalized subsidy is zero at the cap and otherwise ceil((3820000000000000000 − S) / 4194304). The first non-genesis block at full weight allowance pays 91.075893023 Noticoin before fees. Between the effective reference weight and twice that weight, the subsidy is reduced by the block weight penalty. A 0.3 Noticoin minimum reference enters the fee calculation independently of future issuance.

Cumulative issuance as a share of the cap



Illustrative model at 45 seconds per block and 365.25 days per year. Values approximate full subsidies before block weight penalties and atomic-unit rounding.

Modeled block subsidy over time



Values are Noticoin per block before transaction fees, weight penalties and atomic rounding.

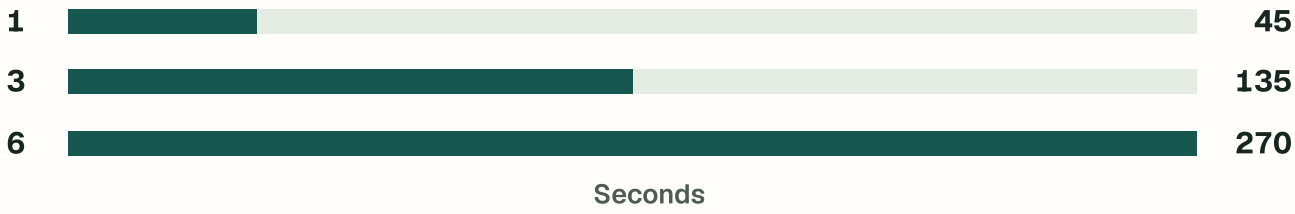
04

Confirmations and spendability

After height 160, ordinary outputs become spendable after six confirmations. Mined outputs mature after 160 blocks. Automatic note delivery checks three confirmations and valid key shares. Actual waiting times vary with inclusion, block discovery and possible chain reorganizations.

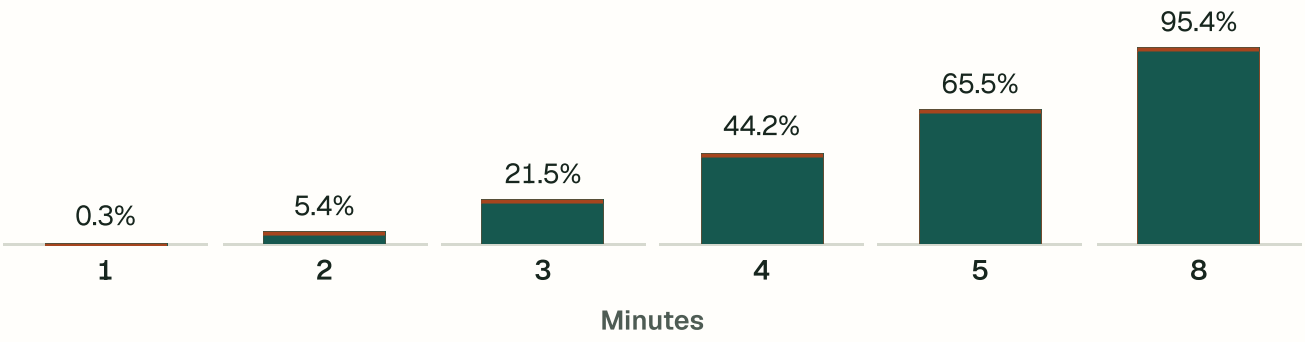
Use the containing block to count confirmation one. A pending transfer in the mempool has no confirmation. Before version 17 the ordinary consensus age is 27 blocks. A height-based rule measures blocks rather than seconds. The elapsed time estimates on this page start from an assumed inclusion in the first arrival and exclude congestion, difficulty changes and reorganizations.

Target time by number of confirmations



Estimates use the 45-second block target.

Modeled probability of six block arrivals



A Poisson arrival model assumes independent finds and a stable mean of 45 seconds. The probability is the chance of at least six arrivals by the time on the axis.

05

Wallet and transaction privacy

RingCT hides amounts, and one-time outputs conceal the recipient address on the blockchain. You can create subaddresses for different contacts.

A wallet scans outputs using its private view key and signs spends with spend-key material. The chain records output keys, input rings, key images, commitments and proofs. For ordinary RingCT inputs under the current rules, the ring covers one real member and fifteen decoys. CLSAG proves authorization over the ring, while a key image prevents an input from being spent twice. Bulletproofs+ proves amounts lie in the allowed range. The public fee and transaction weight remain observable.

Use a distinct subaddress when you need separate receiving contexts. A wallet backup covers its recovery phrase and, separately, any local map profile needed for private pins or purchases. A blockchain observer can see timing and transaction structure. Network peers and services can observe some connection metadata.

Field	Scope
RingCT	Confidential amount commitments
CLSAG	Linkable ring authorization
Bulletproofs+	Proof that amounts are in range
Public transaction data	Fee, weight, structure and timing

06

Map record format and lifecycle

With chain anchors enabled, the wallet embeds a signed 118-byte record in a tx_extra nonce. The signature covers bytes 0 through 53. The map verifies the signature and three confirmations before applying a publication or removal. The public record anchors the pin ID and its author key. The map service handles expiry, title, coordinates and ciphertext. A free publication or removal sends one atomic unit to the author's own wallet and consumes a network fee.

Byte offsets	Field	Value
0–4	Magic and version	`4e544d5001`
5	Action	1 · Publish a pin 2 · Removal
6–21	Pin UUID without separators	16 B
22–53	Ed25519 author public key	32 B
54–117	Ed25519 signature	64 B

Public notes use AES-256-GCM and publish the reading key with the ciphertext. Shared pins stay in the sender's wallet until explicitly packaged for a recipient using RSA-OAEP with SHA-256. The receiving wallet opens the package locally. The Atlas separates map tiles from note storage, so publishing a pin does not update the underlying tiles.

07

Operator registry and epochs

Operators announce a network, expiration height, Ed25519 identity, RSA-3072 encryption key and HTTPS endpoint in a signed transaction record. The node computes an epoch roster from canonical blocks through the cutoff, six blocks before the boundary. It ranks eligible registrations by transaction fee, then by ascending signing identity. Encryption keys and endpoints are deduplicated. The roster requires at least three entries and a strict majority releases a note.

Parameter	Public networks	Regtest
Epoch length	40	10
Maximum registration lifetime	240	30
Cutoff depth	6	6
Roster limit before / from height 200	3 / 10	3 / 10

A purchase uses the roster for the publication block. A newer roster applies to future publications, so older shares must remain reachable for outstanding purchases. Selection proves a registration and a chain position. Operator independence and secure key custody are separate operational assumptions. If too few original operators remain reachable, automatic delivery can stall.

`get_note_operator_roster · NOTI_NOTE_REGISTRY=1`

08

Map records and note release

The author’s wallet encrypts the note with AES-256-GCM before sending it to the map server.

The map server stores the encrypted text of the note. The blockchain records its publication and the payment transaction.

For a paid edition the seller sets its price and a limit of 1–100 copies. The wallet creates a different seller subaddress per copy and publishes a signed offer. A buyer reserves a copy and pays that address. Operators check proof of payment, order binding and three confirmations before returning shares encrypted for the buyer. The buyer reconstructs the AES key locally. The seller’s wallet can be offline after publication when operator delivery is enabled.

Paid note delivery sequence

Signed offer	Reserve a copy	Verify payment	Release key shares	Decrypt locally

The service provides a 24-hour retrieval window after sale. Once the buyer archives a readable copy locally, it can be reopened from Purchased notes until the local copy is removed. A sold-out or expired listing disappears from the public map. The system controls its own delivery path. A reader who opens a note can still save or share its content.

09

Fees and the fund

A publication charge is collected on a dedicated fund wallet subaddress. A separate bridge checks the daemon and wallet identity, chain height and received amount. The website reports balance, available balance and six-confirmation history from snapshots. The snapshot links public transaction hashes to Explorer and can include expense amounts and purpose. Incoming individual amounts and payer addresses are omitted from fund reporting. Fees paid to miners belong to the blockchain transaction and are calculated separately.

Charges by action

Action	Value	Recipient
Public pin, 7 days, standard icon	Network fee	Miner
Paid note listing, 7 days	0.2 Noticoin	Fund
30-day placement	0.6 Noticoin	Fund
Diamond icon	0.1 Noticoin	Fund
Secret note	Set by the author	Author
Pins with donations	Chosen by the sender	Author

Placement and icon fees are combined into one payment to the fund. The sender pays the network fee on top of that amount, and the miner receives it.

10

Mining and pool accounting

Both Stratum modes use RandomX. Shared mining distributes eligible block rewards in proportion to work in the last 128 shares. A solo block pays its miner directly. Pool hashrate uses accepted shares over a ten-minute window. Estimated network hashrate divides difficulty by the block target. Individual miner statistics require the full address.

A Stratum job carries the hashing blob, seed hash, height, share target and job ID. A share proves work against the share target. A block candidate also meets network difficulty and is submitted to the daemon. The shared ledger preserves the last-128-share distribution attached to a canonical candidate. A shared payout is sent once the 160-block reward matures and deducts its transaction fee. In solo mode the daemon builds the miner transaction for the submitting address.

The dashboard uses accepted work over a 600-second window, shortened during startup. It samples the shared and solo estimates and network difficulty every 30 seconds and keeps up to 240 samples for the running pool process. A restart clears chart samples. The payout and share ledger persists independently. The public dashboard exposes aggregate activity, and individual lookup requires the full miner address.

Parameter	Value
PPLNS	128
Hashrate window	600 s
Chart interval and maximum points	30 s · 240
Reward maturity	160 Blocks

11

Verification and launch criteria

The recorded development runs exercise real daemon and wallet binaries with isolated data. A fixed-difficulty regtest is suitable for reproducing transaction and accounting transitions. Public deployment has additional checkpoints for independent peers, sustained hashrate and recovery from network interruption.

Check	Recorded coverage
Consensus and network identity	Genesis and first RandomX block on mainnet, testnet and stagenet in isolated runs
Confirmations and spendability	Wallet relay, peer visibility and six-confirmation spending after activation
Map record format and lifecycle	Author removal, paid delivery and copy limits, direct donation and fund receipt
Mining and pool accounting	Accepted Shared and Solo blocks, ledger restart and confirmed payout

`verify-first-blocks.mjs` · `verify-economy.mjs` · `verify-note-registry.mjs` · `verify-offline-release.mjs` · `pool/tests/integration.mjs`

Release readiness

12

Implementation and operational guides

The Windows CLI bundle includes START-NODE.cmd. Run the node, verify the chain identity and wait for synchronization before checking a balance. Keep administrative RPC on a trusted interface. Use separate node data and wallet files for each network. Verify the release checksum and store the recovery phrase separately from the wallet file.

In a public deployment, set the expected network for every payment-facing component and compare its height-zero hash. Keep Wallet RPC on loopback with authentication. Expose the limited bridge through HTTPS only where the map needs it. Back up the receiving wallet together with the invoice database. Protect operator keys and keep original delivery shares reachable until outstanding retrieval windows end. Use separate data directories for tests and public services.

Component	Operational interfaces
Daemon RPC	Chain identity, height and mining templates `get_info, get_block_header_by_height, get_block_template`
Wallet RPC	Balance, address and transfer proof `get_balance, get_address, check_tx_proof`
Operator registry and epochs	Epoch roster with signing identities and release keys `get_note_operator_roster`
Bridge	Limited payment checks and subaddress allocation
Pool HTTP	Aggregate statistics plus an address-based POST lookup `GET /api/stats · POST /api/miner`

- Network manifest (/assets/noticoin-network.json)
- Desktop wallet guide (/guide.html) · Run a node (/guide-node.html) · CLI & Wallet RPC (/guide-cli.html)
- Map and pins (/map.html) · Payments (/payments.html) · Privacy in Noticoin (/privacy.html)
- Mining and pool (/mining.html) · Fund report (/fund.html)