



NIGHTFALLCOIN

ENGLISH EDITION / SEPTEMBER 2026

Private settlement. Verifiable supply.

A technical whitepaper on confidential payments, monetary accounting and the limits of the current implementation.

WHITEPAPER

A native privacy currency.

Confidential values. One-time outputs. Explicit monetary rules.

0

Premine

90M

Supply ceiling

15s

Target block time

Release 0.9.5 / Protocol 8 / Wire 6

Independent review outstanding. Mainnet swaps disabled.



A READING MAP

The design. The evidence. The limits.

Start with the thesis and trust boundaries. Use the later chapters for implementation detail, risk analysis and source references.

01	The case for private settlement	03
02	One currency, explicit trust boundaries	04
03	From an address to a one-time output	05
04	A compact cryptographic foundation	06
05	Confidential value, public accounting	07
06	Proof of work, with bounded finality	08
07	A monetary schedule written in integers	09
08	Every era, down to the last dark	10
09	Privacy is a threat model, not a slogan	11
10	Disclosure belongs to the holder	12
11	Self-custody across two interfaces	13
12	Propagation without pretending it is anonymity	14
13	Synchronization is part of the trust model	15
14	Atomic swaps: research behind a closed gate	16
15	Evidence with a defined scope	17
16	The risks that govern deployment	18
17	Minimal authority, visible responsibility	19
18	Progress measured by acceptance criteria	20
19	Implementation reference	21
20	Sources and document scope	22

This document separates implemented behavior, known limitations and proposed research. It does not announce an investment product or certify security.

01 / THESIS

The case for private settlement

A payment system should verify the transfer of value without making every user's financial history public.

NIGHTFALLCOIN (NIGHT) is a native proof-of-work currency built around confidential values, non-interactive stealth payments and publicly checkable monetary accounting. Its ledger represents amounts as commitments, authorizes spends with one-time keys and aggregates transaction components into canonically ordered blocks. It is a settlement system, not a general-purpose smart-contract platform.

The central design choice is to combine private payment amounts with an explicit supply invariant. Nodes check a relationship between unspent commitments, accumulated kernel excesses, issuance and burned fees. That relationship is useful only together with valid range proofs, signatures, authorization and correct state transitions. An equation by itself is not a security proof.

Confidential by construction

Native transfer amounts and recipient addresses are not published in cleartext.

Accountable monetary rules

Integer issuance, a fixed ceiling and public fee accounting are enforced by the client.

User-controlled disclosure

View keys and payment receipts expose different information without granting spending authority.

This paper explains the implementation represented by release 0.9.5, protocol 8 and wire version 6. It also identifies operational trust, unresolved security risks and research objectives. It makes no claim of universal anonymity, hardware neutrality, guaranteed finality or investment performance.

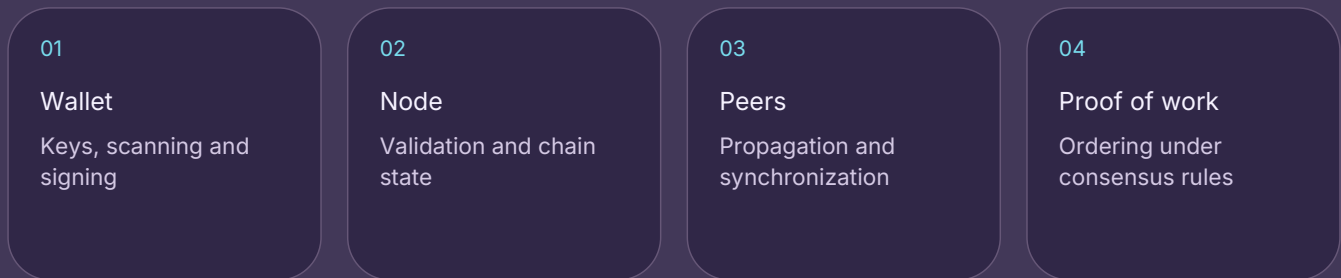
Mainnet exists. Assurance is still developing.

No independent audit is established by this paper. Mainnet atomic swaps remain disabled. Experimental swap recovery can leave NIGHT permanently locked. These are present limitations, not footnotes to a launch promise.

02 / SYSTEM ARCHITECTURE

One currency, explicit trust boundaries

The wallet, the node and the website perform different jobs. Their security claims should not be interchangeable.



A wallet constructs outputs and authorizes spending. A full node maintains the unspent-output set, validates candidate blocks and selects a permitted history by accumulated work. Peers deliver data; their messages are inputs to validation, not authority to create money. Miners assemble blocks and search for an acceptable proof of work.

Core and the browser are not equivalent validators

Core combines a local node, wallet and miner. The browser wallet performs key operations locally but obtains chain data through a remote light API. A remote node can withhold or misreport information even though node access alone does not reveal the spending key. A compromised wallet application or device is a separate, stronger threat.

Boundary	What crosses it	What must not be inferred
Wallet to node	Queries and signed transactions	Acceptance is not confirmation.
Peer to full node	Headers, blocks and transactions	A peer's claims are not validation.
Browser to website	HTTPS requests and application code	HTTPS does not make first-party code trustworthy.
Website to light node	Proxied requests over the configured transport	A displayed balance is not an independently verified chain.

Consensus rules do not remove implementation or distribution trust. A user still chooses a client binary, its checkpoint policy, an operating system and a way to obtain peers. These dependencies are made explicit throughout this paper.

03 / PAYMENT CONSTRUCTION

From an address to a one-time output

Receiving a payment does not require the recipient to participate in an interactive signing session.

The address contains a scan public key A and a spend public key B . A sender samples an ephemeral scalar r and publishes $Ke = rG$. The sender and receiver derive the same shared secret from rA and aKe , where a is the scan secret. Domain-separated derivations produce a blinding scalar, a one-time key offset and payload-encryption material.

$$\begin{aligned} A &= aG \quad B = sG \quad Ke = rG \\ \text{shared} &= KDF(rA) = KDF(aKe) \\ Ko &= B + oG \quad C = vG + \text{rho} H \end{aligned}$$

Conceptual notation: a is the scan secret; s the spend secret; o a derived offset; rho the commitment blinding scalar. Exact domains and byte encodings are defined in code.

The output carries a commitment, range proof, ephemeral key, one-time public key, view tag, encrypted payload and sender signature. The receiver scans for matching outputs, decrypts the value and memo, and checks that the recovered opening recreates the published commitment. A successfully decrypted payload alone is not enough.

Why the sender cannot reclaim the payment

The sender knows the derived offset and blinding factor, but not the recipient's spend secret s . Spending requires authorization under Ko , whose secret is $s + o$. The input verifier obtains Ko from the existing UTXO entry, not from a replacement key supplied by the spender.

A view tag is a scanning hint, not an identity.

A short tag helps reject non-matching outputs before more work is performed. It neither replaces commitment checks nor proves ownership. Fresh ephemeral randomness is essential; this construction must not be implemented by reusing r .

04 / PRIMITIVES AND COMPOSITION

A compact cryptographic foundation

Established primitives reduce the invention surface. Their integration still requires its own security analysis.

Component	Native protocol choice	Role
Group	Ristretto over Curve25519	Prime-order group abstraction and canonical encodings.
Commitment	Pedersen: $C = vG + \rho H$	Hide values while permitting additive accounting.
Range proof	64-bit Bulletproof per output	Constrain the committed value to the supported non-negative range.
Signatures	Generator-parameterized Schnorr	Authorize inputs and bind kernel/output fields.
Hashing	Domain-separated BLAKE3	Separate cryptographic messages and derivation purposes.
Payload	XChaCha20-Poly1305	Authenticated encryption of value, blinding and padded memo.
Mining	Argon2id-based Nighthash-v2	Memory-hard proof-of-work computation.

Commitments and range proofs use the same generator pair from PedersenGens. Binding relies on the assumed hardness of finding the discrete-log relation between G and H . Proof and signature checks must reject invalid encodings and bind the intended context. Bulletproofs require no trusted setup, but that property does not certify Nightfall's application of them.

Native payment cryptography is distinct from the experimental Bitcoin swap adapter, which also involves secp256k1 and a cross-curve proof. Describing the entire project as a single-curve system would conceal that additional review surface.

Security is a composition question.

Randomness, key derivation, transcript binding, integer arithmetic, message parsing and persistence can invalidate a system assembled from individually sound primitives. This paper is explanatory documentation, not a formal proof of that composition.

05 / THE SUPPLY INVARIANT

Confidential value, public accounting

The ledger checks conservation without publishing each user's balance.

$$\begin{aligned} &\text{sum(outputs)} - \text{sum(inputs)} \\ &+ \text{fee } G - \text{reward } G = \text{sum(kernel excesses)} \end{aligned}$$

Local balance equation. The reward term includes the permitted coinbase value; after subsidy ends, coinbase value is funded by fees rather than new issuance.

For an ordinary transfer, output value plus the public fee equals input value. Its remaining commitment difference is a blinding excess. Kernels carry Schnorr signatures relative to H . The verifier requires both the expected excess and valid signatures; merely publishing the publicly computable difference would prove nothing.

$$\begin{aligned} &\text{sum(UTXO commitments)} - \text{sum(all kernel excesses)} \\ &= (\text{total minted} - \text{total burned fees}) G \end{aligned}$$

Global invariant. UTXO commitments and kernel excesses are group elements; minted and burned values are integer accounting totals.

Starting from the defined genesis state, accepted transitions preserve this invariant. Range proofs exclude invalid value encodings; input signatures establish spending authority; UTXO checks prevent double spending; emission rules bound coinbase issuance. Removing any of these checks changes the security argument.

What the public website can demonstrate

The network page displays a node's reported totals and invariant result. The browser does not independently replay the ledger or verify the full cryptographic statement. Independent verification requires a validating implementation and an understood policy for trusted local state and historical checkpoints.

The lesson from the discarded v4 construction

A comparison against a value recomputed from the same public inputs was not evidence of conservation. The current construction adds the necessary knowledge checks and regression coverage. That history is a reason to demand scrutiny, not a claim that future defects are impossible.

06 / ORDERING AND VALIDATION

Proof of work, with bounded finality

Valid history is selected by cumulative work within the rules enforced by the client.

Mainnet uses Nighthash-v2 with Argon2id parameters of 32 MiB, one iteration and one lane. The design imposes a memory cost on each hashing instance. It does not establish that GPUs or purpose-built hardware have no advantage, that every CPU earns equally, or that mining is profitable. No hardware-efficiency benchmark is claimed here.

Rule	Current mainnet parameter
Target interval	15 seconds; a target, not a delivery guarantee
Retarget	Every block, LWMA-1 over 90 blocks
Difficulty floor	2,000
Timestamp	Above median of previous 11; no more than 120 seconds into the future
Coinbase maturity	1,440 blocks; approximately 6 hours at target spacing
Rewind bound	500 blocks of the node's current history
Compiled checkpoint	Height 25,000 in the reviewed release

A miner's proposed block must satisfy shape, ordering, parent linkage, timing, difficulty, proof-of-work, authorization, commitment and issuance checks. Rejected transitions must not partially alter ledger state. Transaction selection in the current mempool sorts candidates by fee and skips conflicts; fees are nevertheless burned during subsidy eras.

A checkpoint is a trust decision, not finality magic

The client refuses histories conflicting with its compiled checkpoint and rewinds beyond its bound. These rules constrain which forks it can adopt; they do not prevent an adversary from mining a competing history or guarantee automatic recovery from a deep disagreement. Checkpoint-anchored replay can also skip costly verification of a historical prefix.

Confirmations reduce uncertainty; they do not eliminate it.

Reorganization exposure depends on competing work, connectivity, client policy and the value at risk. Fifteen-second blocks should not be presented as fifteen-second irreversible settlement.

07 / ISSUANCE AND FEES

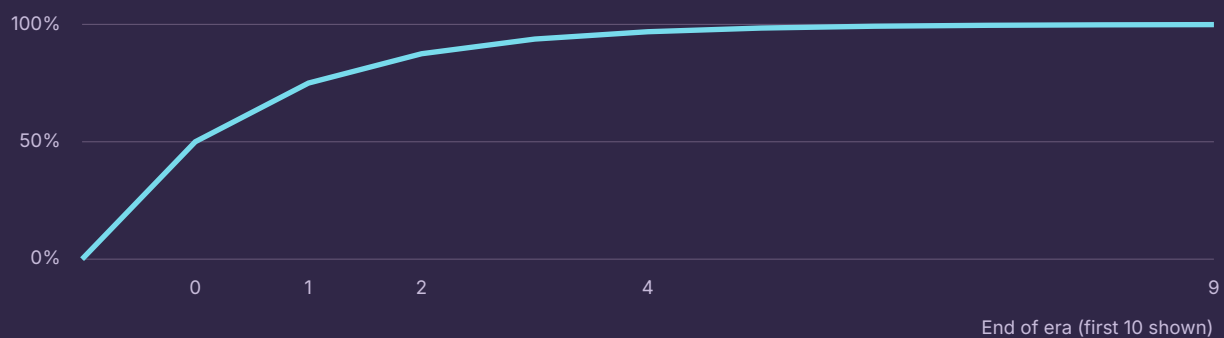
A monetary schedule written in integers

No premine, no tail emission, and a ceiling that the integer schedule never quite reaches.

```
era(h) = floor(h / 7,500,000)
subsidy(h) = floor(600,000,000 / 2^era(h)) darks
```

Height h is zero-based. One NIGHT is 100,000,000 darks. The implementation also enforces the remaining supply ceiling.

Cumulative scheduled issuance



Share of the 90 million ceiling at the end of each era. These are supply shares, not returns. Heights determine issuance; calendar dates do not.

The initial subsidy is 6 NIGHT per block. Each paying era contains 7,500,000 blocks. Integer truncation eventually reduces the subsidy to zero: there are 30 paying eras, with total scheduled issuance of 89,999,999.25 NIGHT. The 90,000,000 ceiling is therefore not a promised terminal balance.

Period	New issuance	Treatment of fees
Subsidy remains positive	Scheduled subsidy to the miner	Fees are destroyed; circulation decreases by the burned amount.
Subsidy is zero	None	Fees fund the miner's coinbase; no additional issuance or fee burn.

A fair-launch rule does not guarantee a fair distribution of ownership. Access to hardware, energy, information and early participation still matters. Likewise, a fee-only future does not guarantee an adequate security budget: demand for settlement and the economics of securing it remain open empirical questions.

08 / EXACT EMISSION SCHEDULE

Every era, down to the last dark

All figures are derived using integer darks, not floating-point approximations of a geometric curve.

Era	First block	NIGHT / block	Cumulative NIGHT
0	0	6	45,000,000
1	7,500,000	3	67,500,000
2	15,000,000	1.5	78,750,000
3	22,500,000	0.75	84,375,000
4	30,000,000	0.375	87,187,500
5	37,500,000	0.1875	88,593,750
6	45,000,000	0.09375	89,296,875
7	52,500,000	0.046875	89,648,437.5
8	60,000,000	0.0234375	89,824,218.75
9	67,500,000	0.01171875	89,912,109.375
10	75,000,000	0.00585937	89,956,054.65
11	82,500,000	0.00292968	89,978,027.25
12	90,000,000	0.00146484	89,989,013.55
13	97,500,000	0.00073242	89,994,506.7
14	105,000,000	0.00036621	89,997,253.275
15	112,500,000	0.0001831	89,998,626.525
16	120,000,000	0.00009155	89,999,313.15
17	127,500,000	0.00004577	89,999,656.425
18	135,000,000	0.00002288	89,999,828.025
19	142,500,000	0.00001144	89,999,913.825
20	150,000,000	0.00000572	89,999,956.725
21	157,500,000	0.00000286	89,999,978.175
22	165,000,000	0.00000143	89,999,988.9
23	172,500,000	0.00000071	89,999,994.225
24	180,000,000	0.00000035	89,999,996.85
25	187,500,000	0.00000017	89,999,998.125
26	195,000,000	0.00000008	89,999,998.725
27	202,500,000	0.00000004	89,999,999.025
28	210,000,000	0.00000002	89,999,999.175
29	217,500,000	0.00000001	89,999,999.25

Era e covers heights $e \times 7,500,000$ through $(e + 1) \times 7,500,000 - 1$. At height 225,000,000 the scheduled subsidy is zero. Fees burned in earlier eras are not reissued. Actual circulating supply can therefore be lower than the cumulative issuance shown.

09 / WHAT OBSERVERS CAN LEARN

Privacy is a threat model, not a slogan

Hiding amounts and addresses is meaningful. It is not the same as erasing every relationship or concealing every network interaction.

Observer	Protected information	Remaining exposure
Passive chain reader	Plaintext transfer values and reusable recipient addresses	Input/output structure, public fees, block timing, coinbase flags and graph information.
Payment counterparty	Unrelated private keys and undisclosed wallet history	Its own payment, the address shared with it and out-of-band context.
First-hop peer or network observer	Some origin information can be obscured by relay/proxy choices	Timing, topology, fallback paths and correlation across observations.
Light-API operator	Spending key is not required for serving data	Requests, submission timing and the ability to misreport or withhold chain data.
Compromised wallet device	No general protection established	Keys, decrypted history, signing actions and local backups can be exposed.

Aggregation is not graph erasure

Blocks merge and sort inputs, outputs and kernels rather than retaining their original transaction grouping. Observers who saw transactions before aggregation may retain that grouping. Low activity also limits the ambiguity a block can provide. A block containing a single payment is not a large anonymity set.

Cut-through is not applied in this construction. Inputs retain explicit authorization under one-time output keys. Simply removing created-and-spent components would discard evidence needed by the present validation design. This is a constraint of this implementation, not a proof that every future non-interactive design must make the same trade-off.

No universal anonymity claim

Nightfall does not claim to defeat a global observer, malicious endpoint software or disclosure by a counterparty. No numeric anonymity score or measured resistance to deanonymization is established by this paper.

Disclosure belongs to the holder

Proving one payment should not require handing over the ability to spend or the history of every payment.

Address / nf1

A destination shared with a sender. Not a spending credential.

View key / nfview1

Scanning and decryption authority for matching outputs. Financially sensitive, but not a spend key.

Payment receipt

An opening and signature for one output. A narrower statement than sharing a view key.

A view key contains the scan secret and spend public key. It enables the holder to identify and open matching outputs without the spend secret. Treat it as continuing access to sensitive financial information, including future matching outputs. It is not a harmless public identifier or a revocable web-session token.

A receipt discloses an individual commitment opening and a signature by the address's spend key. Verification checks the opening and signature. A verifier still needs appropriate chain evidence to establish inclusion and confirmation; the receipt alone does not establish that the output remains unspent.

01

Choose the statement

One output or broader viewing

02

Verify cryptography

Opening and signature

03

Check chain context

Inclusion and confirmations

A recipient should ask what the verifier actually needs before sharing a view key. A receipt can support reconciliation or evidence of a specific transfer; it is not a legal attestation, identity certificate or guarantee of exclusive ownership. Time-window proofs and balance lower bounds remain research objectives, not current receipt features.

Never disclose recovery words for verification.

A request for proof of payment never justifies sharing the spend seed. Recovery words are not an alternative receipt format.

Self-custody across two interfaces

The same asset is available through different operating models, with different failure modes.

	Core / desktop	Web / desktop and mobile
Chain data	Local node and stored chain state	Remote light API
Key operations	Local application	Local browser application / WebAssembly
Mining	Integrated	Not a browser mining client
State retention	Application data directory	Browser storage can be cleared or lost
Mainnet swaps	Disabled	Not offered
Main dependency	Device, client distribution and validation policy	Those risks plus first-party scripts and remote display trust

Recovery words are a specific format

Nightfall encodes its 32-byte wallet seed directly as 24 BIP-39 words. It does not derive that seed using BIP-39's PBKDF2 wallet-seed procedure, and it does not support a BIP-39 passphrase or '25th word'. The familiar word encoding does not imply Bitcoin or Ethereum wallet compatibility. Do not reuse another wallet's phrase.

The phrase recovers key material, not every piece of application state. Contacts, preferences and independently generated experimental swap secrets require separate treatment. In particular, the 24 words cannot reconstruct a lost swap-secret file after funds have been locked.

Browser storage is not a secure backup.

The current web wallet persists exported seed-containing state in localStorage. The website and wallet share an origin. Malicious first-party JavaScript, an extension or a compromised device can expose that state. Separating origins and protecting locked storage are priorities, not shipped guarantees.

Wallet interfaces must distinguish spendable, pending and immature funds. A retry is a rebroadcast attempt, not evidence of confirmation. Back up recovery words offline, protect the device and verify the destination before signing.

12 / NETWORK AND LIGHT ACCESS

Propagation without pretending it is anonymity

Privacy, availability and resource limits must be evaluated together.

Native peers exchange bounded newline-delimited JSON messages over TCP. Handshakes check the network, wire version and genesis. The implementation caps individual P2P messages at 4 MiB and serialized blocks at 2 MiB. These ceilings are defenses against some resource-exhaustion paths, not a complete denial-of-service analysis.

Dandelion-class relay, accurately described

A locally originated transaction initially takes a stem path. Relays probabilistically continue stemming, with a configured probability of 90%, or fluff to a wider set. The current session-based path records an embargo of 12-28 seconds and later fluffs transactions still retained in the mempool. A fallback path for absent live sessions has different delivery behavior.

These mechanisms are inspired by Dandelion research; the paper's formal guarantees cannot simply be assigned to this implementation. Topology, adversarial peers, timing and fallback behavior all matter. Delivery still depends on connectivity, and pending-send retry logic remains relevant.

Transport boundary	Current behavior	Limitation
Native outbound peers	SOCKS5/Tor configured by default	Unavailable proxy can lead to cleartext fallback; .onion destinations do not use that fallback.
Browser to website	HTTPS and same-origin proxy	Does not hide requests from the website operator.
Worker to light upstream	One configured HTTP upstream	No upstream TLS or independent failover established in this configuration.

Decentralized rules are not decentralized operations.

Independent validation is valuable, but discovery services, hosting and release distribution can remain operational bottlenecks. Multiple names resolving to the same machine do not create independent redundancy.

13 / STORAGE AND BOOTSTRAP

Synchronization is part of the trust model

Faster delivery of chain data should not be confused with stronger evidence that the data is correct.

The client supports binary chain storage as well as legacy JSON storage. A disk-format change is not a consensus change: block identity is not defined by the choice between those encodings. Operating-system data-directory locks prevent concurrent writers; they neither encrypt keys nor defend against a compromised local account.

01

Obtain archive

Versioned file and manifest

02

Check integrity

Size and published checksum

03

Import and validate

Client rules and replay policy

04

Catch up

Compare and synchronize with peers

Bootstrap archives reduce the work of obtaining historical bytes. A checksum confirms agreement with the published checksum, not the honesty of an operator who controls both. The archive is not an independent trust anchor. Its contents remain subject to the client's import and replay policy.

Trusted replay must be named

Local validation records and checkpoint-anchored replay can avoid repeating expensive checks. The checkpoint optimization requires a reached mainnet pin with the expected hash.

NIGHTFALL_NO_ASSUME_VALID disables that checkpoint-based optimization, not every use of previously validated local state. 'Every proof is checked on every startup' would be inaccurate.

Pruned operation retains the state and recent history needed by the supported local policy, including the last 500 block bodies. It cannot replace an archival peer for all historical or light-API requests. If the evidence needed to trust pruned local state is invalid, omitted history may need to be obtained again.

Recovery procedures are a security feature.

Operators need a documented response to corrupted storage, deep forks, conflicting checkpoints and unavailable archives. Neither a successful download nor a faster progress bar demonstrates independently validated settlement.

Atomic swaps: research behind a closed gate

NIGHT/BTC exchange is experimental. Mainnet wallets must not treat the existence of swap code as permission to lock real funds.

The experimental protocol coordinates Bitcoin P2WSH 2-of-2 transactions, ECDSA adaptor signatures, relative-timelock abort paths and a shared NIGHT spend condition. A cross-curve proof connects key material across the two cryptographic domains. Participants exchange protocol packets directly; no custodial bridge is required by this design.

Outcome	Bitcoin side	NIGHT side
Successful exchange	Alice redeems as intended	Bob obtains the required material to claim.
Bitcoin refund published	Bob recovers through the abort path	Alice's NIGHT recovery can depend on the revealed share.
Counterparty never refunds	A punishment path may exist	NIGHT can remain permanently locked.
Late redeem or reorganization	A redeem may not survive or confirm	A disclosed secret cannot be made secret again.

No independent timed NIGHT refund

Conditional recovery after a Bitcoin refund is not a NIGHT timeout that Alice can enforce alone. Punishing Bitcoin behavior does not unlock NIGHT. This loss mode is why the word 'atomic' must not be read as a guarantee of cost-free abort or guaranteed capital recovery.

The withdrawn refund proposal treated a co-signed input as if it committed to a complete future transaction. It does not: input authorization and kernel fields have separate bindings so aggregation can work. A Bitcoin-style pre-signed transaction intuition cannot be imported without proving that the destination chain enforces the intended condition.

Current hardening includes persisted intent before broadcast, exact outgoing packet recovery and refusal of unsafe state transitions. External review of the cross-curve construction alone is not sufficient to remove the permanent-lock issue. Any future mainnet decision requires protocol, loss-model, recovery and implementation review together.

15 / VERIFICATION AND REVIEW

Evidence with a defined scope

A useful assurance statement names what was tested, against which source, and what was not covered.

277

Targeted tests passed

9

Explicitly ignored

0

Failures in that run

The internal review dated 8 September 2026 records a targeted Rust 1.98.0 test run executed on 7 September across ledger, consensus, storage, wallet and swap packages. The totals above are evidence from that review, not a fresh full-workspace run performed for this whitepaper.

Evidence	Examples of covered behavior
Ledger regressions	Thin-air mint rejection, authorization, invalid values and payload tampering.
Consensus regressions	Emission bounds, fork choice, checkpoint mismatch and rejection without state mutation.
Storage and wallet	Directory locking, foreign validation records, reorg persistence and output reservation.
Swap state tests	Persist-before-send, handshake restart and refusal of unsafe redeem/cancel transitions.

The nine ignored cases include a performance measurement, a long soak and Bitcoin-node-dependent integration tests. They are not counted as passes. The review did not establish a fresh Windows runtime assessment, external cryptographic proof, full dependency/CVE audit, server penetration test or active mainnet attack analysis.

Native implementation claims are anchored to v0.9.5 and selected code paths were checked against outdated prose. Website configuration and the September review are separately dated; the live site is not part of that release baseline.

Independent review remains outstanding.

This is an AI-assisted editorial and technical synthesis. It is not an independent audit, a security certification or a guarantee against unknown defects. Reproducible tests support specific claims; they cannot justify a blanket '100% safe' label.

16 / SECURITY AND ECONOMIC LIMITATIONS

The risks that govern deployment

Read these as present constraints on use, not as problems a future roadmap has already solved.

Risk	Consequence	Required response
Cryptographic or implementation flaw	Loss, unauthorized spend or invalid issuance may be possible despite passing tests.	Independent specialist review and adversarial regression work.
Browser origin or device compromise	Seed-containing state or signing actions can be exposed.	Protect devices; separate wallet origin and assess locked-storage protection.
Low or concentrated mining work	Reorganizations, censorship or stalled settlement may become cheaper.	Measure distribution and stress-test adverse network conditions.
Checkpoint or deep-fork disagreement	Nodes can refuse a history or diverge rather than recover automatically.	Document checkpoint policy and operator recovery procedures.
Light-node and hosting concentration	Misleading balances, request visibility or service interruption.	Independent upstreams, authenticated transport and clearer client verification.
Experimental swap failure	Permanent capital lock or asymmetric loss after secret disclosure.	Keep mainnet gated; review the complete loss and recovery model.
Unsigned or compromised distribution	An attacker can replace software and the checksum alongside it.	Publisher authentication, reproducibility work and safer release provenance.
Future fee-only security budget	Settlement demand may not support sufficient mining expenditure.	Analyze economics without promising price, adoption or yield.

This is not an exhaustive risk inventory. No token price, exchange listing, liquidity, investment return or adoption forecast is asserted. Protocol rules and user demand cannot guarantee each other. Users should not infer suitability for a particular financial or operational purpose from the visual polish of a wallet or document.

Minimal authority, visible responsibility

No monetary administrator does not mean no human influence. Software and operational decisions still need accountability.

The current monetary construction has no premine, special treasury allocation or administrative mint/freeze credential. That is narrower than saying nobody can influence the system. Maintainers choose releases, operators run infrastructure, miners choose which transactions to include and users choose which software and chain rules to accept.

Protect the monetary rules

Do not present discretionary issuance, special allocations or price-responsive changes as routine maintenance.

Keep disclosure voluntary

A payment receipt or view key should be a holder's choice, not an invisible prerequisite for native transfers.

Make trust changes explicit

Checkpoint updates, key formats and consensus changes deserve specifications, review and migration plans.

History cannot be redesigned away

Nightfall has discarded earlier chains. The v4 monetary construction was unsound; later iterations exposed engineering and emission-design problems. The current v8 genesis is a distinct chain. Earlier-genesis balances do not automatically migrate into it. This history should remain discoverable and should make future compatibility decisions more deliberate.

In this implementation, protocol-version configuration contributes to genesis identity, while wire version gates network compatibility. A software-version increment does not by itself imply either kind of change. Release documentation must distinguish client updates from chain resets and must not silently reinterpret historical balances.

The intended scope stays narrow.

Native confidential settlement is the focus. This paper does not announce a general-purpose VM, a custodial bridge, wrapped NIGHT, a token sale, an exchange listing or a delivery calendar.

18 / RESEARCH AND ENGINEERING PRIORITIES

Progress measured by acceptance criteria

These are proposed work tracks. None is a promise of a feature, a deadline or a future mainnet activation.

Track	Work to investigate	Evidence before calling it complete
01 / Assurance	External review of value conservation, authorization, key derivation and cross-curve composition.	Published scope, findings, fixes and independent retesting; explicit residual risks.
02 / Custody and distribution	Separate wallet origin, encrypted locked storage and authenticated builds.	Recovery/migration tests, adversarial origin tests and verifiable release provenance.
03 / Network resilience	Independent light nodes, upstream TLS and stronger relay evaluation.	Measured outage recovery, operator diversity and privacy analysis of actual topology/fallbacks.
04 / State and synchronization	Incremental authenticated state structures and more efficient validation.	Documented trust model, benchmark methodology, corruption tests and equivalence to validated state.
05 / Private authorization	Study authorization that permits more pruning without weakening non-interactive ownership.	A construction, security argument and adversarial evaluation before any consensus proposal.
06 / Cross-chain exchange	Resolve or explicitly constrain abort and secret-disclosure loss modes.	End-to-end loss analysis, real-node recovery tests and external review before a mainnet decision.

Performance claims should publish hardware, workload, implementation version and verification cost. Privacy claims should publish the adversary and observation model. A proposed optimization that changes which evidence a node checks is also a trust-policy change, even if the user interface calls it faster sync.

Code, evidence, then a release decision.

A passing milestone does not automatically authorize mainnet activation. Compatibility, operator recovery and user-loss analysis belong to the decision, not to the post-release checklist.

19 / PARAMETERS AND TERMINOLOGY

Implementation reference

A compact reference for readers moving from this paper to code. The tagged implementation remains authoritative for exact behavior.

Parameter	Value in this edition
Native asset / smallest unit	NIGHT / dark; 100,000,000 darks per NIGHT
Software baseline	v0.9.5
Protocol / wire / magic	8 / 6 / NFL2
Scheduled ceiling / terminal issuance	90,000,000 / 89,999,999.25 NIGHT
Initial subsidy / halving interval	6 NIGHT / 7,500,000 blocks
Target block time / coinbase maturity	15 seconds / 1,440 blocks
Mainnet proof of work	Argon2id: 32 MiB, 1 iteration, 1 lane
Block / message byte ceilings	2 MiB / 4 MiB
Local rewind bound / retained pruned bodies	500 / 500 blocks
Mainnet swap availability	Disabled

```
061a052d49607ff8f4b306c75d622ebd
230cff4ec3a45a6dff2f7738d4b20de
```

Mainnet genesis hash, shown across two lines. Concatenate without whitespace when comparing.

UTXO: an unspent output recorded by the ledger. Commitment: a group element representing a hidden value and blinding factor. Kernel: a signed record of public transaction terms and excess. Range proof: evidence that a committed value is within the allowed interval. Coinbase: the miner's permitted output, subject to maturity. Reorganization: replacement of a suffix of accepted history. Checkpoint: a compiled height/hash constraint. Bootstrap: a delivery format for historical chain data, not a new consensus authority.

A view key enables observation, not spending. A receipt proves a limited cryptographic statement, not continuing unspent status. 'Live' denotes a reported or running state; it does not mean independently audited or irreversible.

20 / REPRODUCIBLE REFERENCES

Sources and document scope

Implementation references are pinned to v0.9.5. The website review is dated separately. External literature describes primitives, not an endorsement of Nightfall.

- [01] Protocol identity, monetary constants and PoW parameters
- [02] Native cryptography: commitments, stealth, signatures and mnemonic encoding
- [03] Ledger validation, UTXO authorization and exploit regressions
- [04] Consensus, emission, checkpoints, replay and mempool selection
- [05] Node runtime: session relay, embargo timers and network behavior
- [06] Storage implementation and operating-system directory locks
- [07] Wallet keys, state and receipt implementation
- [08] Experimental swap driver, persistence and mainnet availability gate
- [09] Swap loss model and conditional NIGHT recovery
- [10] Project history and discarded chain epochs
- [11] Internal security review, 8 September 2026 - scope and open risks
- [12] RFC 9496: The ristretto255 and decaf448 Groups (2023)
- [13] Bunz et al., Bulletproofs: Short Proofs for Confidential Transactions and More (2018)
- [14] RFC 9106: Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications (2021)
- [15] Fanti et al., Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees (2018)
- [16] BIP 39: Mnemonic code for generating deterministic keys

Editorial status: English edition, 8 September 2026; AI-assisted source synthesis and design revision. Supersedes the earlier 0.9.2 whitepaper. No independent audit was performed in creating this document. The PDF and HTML edition are generated from the same content source.

A durable standard for the project

Make private settlement understandable. Make monetary rules inspectable. Make every important limitation visible before a user commits funds.