Sovereign

A Fair-Launched, Post-Quantum, Proof-of-Work Chain
Technical White Paper — ticker XUS, network `sov-mainnet`

The Sovereign contributors

Genesis: 00:00 CDT, 4 July 2026 | Genesis hash `cb0272ff...e72d`

Abstract

Sovereign is a fair-launched cryptocurrency with no premine, no founder allocation, and no proof-of-stake: every coin is mined. It secures authorship with a *hybrid* signature (Ed25519 conjoined with ML-DSA-65, FIPS 204) and its peer transport with a hybrid key exchange (X25519 with ML-KEM-768, FIPS 203), so that forging a spend or unmasking a channel requires breaking a classical *and* a lattice assumption simultaneously. Consensus is Nakamoto proof-of-work sealed by RandomX, with a heaviest-cumulative-work fork choice made total by a deterministic tie-break. The monetary base is capped: emission is a halving geometric series summing to a hard ceiling of 21,000,000 XUS. We give explicit theorems and proofs for bounded emission, exponential reversal security, hybrid unforgeability, value conservation (no unauthorized mint), and the computational immutability of genesis. A transparent *account* ledger is paired with an optional Zcash-Orchard shielded pool for unlinkable value, and an over-collateralized CDP stablecoin (XUSD) is analyzed for solvency. We close with an unusually explicit statement of *what this system is not*: it is not FIPS-140 validated, its shielded pool is not post-quantum, and its mainnet has not undergone a third-party audit. Those disclosures are part of the design, not a footnote to it.

Contents

1 Introduction

Most chains ask to be trusted. This one asks to be *checked*. Sovereign was launched at midnight on 4 July 2026 — the 250th anniversary of American independence — with a genesis block whose timestamp cryptographically gated the launch: no valid successor block could be mined before that instant. Genesis carried *zero* supply. There was no sale, no allocation, and no privileged validator set. Authority over the chain is exactly one thing: hashpower.

The design is governed by three commitments.

(i) **Fair issuance.** Every unit is mined under a fixed, halving schedule with a hard cap; the genesis budget arithmetically forbids any pre-allocation (Theorem 5).

(ii) **Honest post-quantum posture.** Where we claim quantum resistance, we prove that breaking the system requires defeating a lattice assumption (Theorem 9); where we *cannot* yet claim it — the shielded pool — we say so plainly (§11).

(iii) **Immutability by construction.** The frozen network identity is enforced by the software itself, not by convention (Theorem 11).

2 Preliminaries and Notation

Let $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{256}$ denote BLAKE3, modeled as a collision-resistant hash and, where noted, a random oracle. Balances are integers in *grains*, with 1 XUS $= 10^8$ g. Time is measured in blocks; the protocol targets a mean inter-block interval $T = 150$ s (2.5 minutes).

Definition 1 (Account and implicit identity)**.** The transparent ledger is a finite map from account identifiers to records $\mathsf{acct} = (\mathsf{nonce} \in \mathbb{N},\ \mathsf{balance} \in \mathbb{N}_{\mathsf{g}},\ \mathsf{key} \in \mathsf{PK} \cup \{\bot\})$. A public key pk controls the *implicit account* $\mathsf{id}(pk) = \mathsf{H}(\mathrm{enc}(pk))$ rendered as lowercase hex, where enc is the canonical Borsh encoding. Thus an account identifier commits to its key under H, and the key itself is revealed only when a transaction is first signed from that account.

This is an *account* model, not a UTXO model: a transfer debits a balance and credits another; there are no transaction outputs and hence no change addresses. Replay is prevented by the strictly increasing per-account nonce.

3 Consensus

3.1 Proof of work

A block header hdr commits to (height, previous hash, transaction root, receipts root, state root, timestamp, proposer, difficulty bits, nonce). The seal is $\sigma = \mathrm{RandomX}(\mathsf{hdr})$ on mainnet (SHA-256d on development networks), and the block is valid only if $\sigma \le \tau(\mathsf{bits})$ where τ is the target decoded

from the compact difficulty. RandomX is memory-hard and CPU-optimized, narrowing the ASIC advantage and keeping issuance broadly distributed. Because the seal is recomputed over the *whole* header at import, every header field — proposer included — is bound by the work; tampering any field invalidates σ.

3.2 Fork choice and finality

Let $W(C) = \sum_{b \in C} 2^{256}/(\tau(b)+1)$ be the cumulative work of chain C. Nodes adopt

$$C^\star = \arg\max_{C} \ \big(W(C),\ -\min_{\text{tie}} \mathsf{H}(b_{\text{tip}})\big),$$

i.e. heaviest cumulative work, with ties broken by the *smaller* tip hash. The tie-break makes the preference a total order that every honest node computes identically, so equal-work competitors converge rather than persisting as a fork. Finality is probabilistic in confirmation depth, quantified next.

Assumption 2 (Honest majority)**.** An adversary controls a fraction q of global hashpower and honest miners control $p = 1 - q$, with $q < p$. Block discovery is a memoryless Poisson process at the network's difficulty.

Theorem 3 (Exponential reversal security)**.** *Under Assumption 2, the probability that the adversary ever replaces a transaction buried under z honest confirmations is at most $(q/p)^z$.*

Proof. Consider the lead of the honest chain over the adversary's private chain as a random walk that steps $+1$ when an honest block is found (probability p) and -1 when an adversary block is found (probability q). A successful reversal is the event that this walk, started at height z, ever reaches 0. This is the classical gambler's-ruin problem on $\{0, 1, 2, \dots\}$ with up-probability p and down-probability q. For $q < p$ the probability of ever hitting 0 from a start of z is $(q/p)^z$: writing u_k for the hitting probability from k, the harmonic relation $u_k = p\,u_{k+1} + q\,u_{k-1}$ with $u_0 = 1$ and $u_k \to 0$ as $k \to \infty$ has the unique bounded solution $u_k = (q/p)^k$. The Poisson timing only rescales the clock and does not change the embedded up/down walk, so the bound $(q/p)^z$ holds. (This is the Nakamoto bound; a finite-time refinement replaces the ceiling with a Poisson catch-up sum but preserves the exponential decay.) □

Corollary 4. *Security is exponential in confirmations: for $q = 0.1$, six confirmations bound reversal by $(1/9)^6 \approx 1.9 \times 10^{-6}$; for $q = 0.3$, by $\approx 7.5 \times 10^{-4}$.*

3.3 Difficulty adjustment

Difficulty retargets every block by a linearly-weighted moving average (LWMA) of recent solve-times over a window of N blocks: more recent intervals receive proportionally greater weight. The next target τ_{n+1} is set so that the expected next interval equals T given the weighted-mean observed rate. The weighting is an unbiased, low-variance estimator of the instantaneous hashrate; its unique fixed point is the solve-time T, so under a constant hashrate the interval process is stationary about T, and under a step change in hashrate it re-converges in $O(N)$ blocks without the oscillation of naive per-block ratio retargets.

4 Monetary Policy

Let $R_0 = 12.5$ XUS be the initial block reward and $H = 840{,}000$ the halving period. The reward at height $h \geq 1$ is

$$R(h) = R_0 \cdot 2^{-\lfloor (h-1)/H \rfloor}.$$

Theorem 5 (Bounded, convergent emission)**.** *Ignoring integer-grain rounding, total ever-issued supply is exactly*

$$S_\infty = \sum_{h=1}^{\infty} R(h) = HR_0 \sum_{i=0}^{\infty} 2^{-i} = 2HR_0 = 2 \cdot 840{,}000 \cdot 12.5 = 21{,}000{,}000 \text{ XUS}.$$

With rewards floored to whole grains each halving epoch, the realized total is $S^\star = 20{,}999{,}999.9076$ XUS $< S_\infty$*. Genesis supply is* 0*.*

Proof. Emission is constant $R_0 2^{-i}$ across each block of the i-th epoch of H blocks, so the epoch contributes $HR_0 2^{-i}$; summing the geometric series with ratio $1/2$ gives $HR_0 \cdot 2$. The cap $2HR_0 = 21{,}000{,}000$ follows. Under grain flooring, each epoch contributes $H \cdot \lfloor R_0 2^{-i} \cdot 10^8 \rfloor / 10^8 \leq HR_0 2^{-i}$, and the geometric tail is finite; evaluating the floored sum yields 20,999,999.9076. Genesis carries no allocation, so the height-0 contribution is 0. □

Remark 6. Because the emission budget is fixed at 21M and the genesis routine credits no accounts, *any* nonzero genesis allocation would push scheduled issuance over the cap and is rejected arithmetically at genesis construction. "No premine" is therefore a checked invariant, not a promise. There is no stake, no slashing, and no committee: a block's coinbase pays its proposer, and hashpower is the only vote.

5 The State Transition and Value Conservation

The ledger state S commits to accounts, contract storage, shielded commitments and nullifiers, tokens, and auxiliary maps under a single H-based state root that each node recomputes and checks against the header. Define the *monetary mass*

$$V(S) \;=\; \sum_{a} \mathsf{balance}(a) \;+\; \mathsf{shieldedValue}(S),$$

counting only the native asset (issued tokens are separate assets and are conserved within their own supply).

Theorem 7 (Value conservation / no unauthorized mint)**.** *Let* $S \xrightarrow{B} S'$ *be the application of a valid block* B *that pays coinbase reward* $R(h)$*. Then* $V(S') = V(S) + R(h)$*, and no transaction in* B *changes* V*.*

Proof. By induction over the block's actions. A *transfer* debits x from one balance and credits x to another: $\Delta V = 0$. A *shield* moves x from a transparent balance into the pool, increasing $\mathsf{shieldedValue}$ by exactly the Orchard bundle's signed value balance x: $\Delta V = 0$; a *de-shield* is the mirror image; a fully shielded transfer has zero value balance: $\Delta V = 0$. Fees are conserved (burned or paid to the proposer, both inside V). No action mints the native asset. The only source of new mass is the coinbase, which adds exactly $R(h)$. Hence $V(S') = V(S) + R(h)$. The node rejects any block whose recomputed state root or value balance violates this identity, so an over-mint cannot be finalized. □

Corollary 8. *Circulating native supply at height* n *equals* $\sum_{h \leq n} R(h)$ *exactly; the invariant* $\sum_a \mathsf{balance}(a) + \mathsf{shieldedValue} = \mathsf{mined}$ *holds at every committed state.*

6 Cryptographic Authorship: the Hybrid Signature

A public key is the pair $pk = (pk_{\mathrm{ed}}, pk_{\mathrm{ml}})$ and a signature is the pair $\sigma = (\sigma_{\mathrm{ed}}, \sigma_{\mathrm{ml}})$; verification is the *conjunction*

$$\mathsf{Vrf}_{pk}(m, \sigma) = 1 \iff \mathsf{Vrf}^{\text{Ed25519}}_{pk_{\mathrm{ed}}}(m, \sigma_{\mathrm{ed}}) \ \wedge \ \mathsf{Vrf}^{\text{ML-DSA-65}}_{pk_{\mathrm{ml}}}(m, \sigma_{\mathrm{ml}}).$$

Both components are *many-time* EUF-CMA signatures (Ed25519; ML-DSA-65 / Dilithium, FIPS 204); neither is a one-time construction, so key reuse across arbitrarily many signatures is sound.

Theorem 9 (Hybrid unforgeability)**.** *Let Σ_{hyb} be the conjunctive scheme above. For any forger $\mathcal{A}$ running in time t,*

$$\mathsf{Adv}^{\textit{euf-cma}}_{\Sigma_{\mathrm{hyb}}}(\mathcal{A}) \ \le \ \min\Big(\mathsf{Adv}^{\textit{euf-cma}}_{\text{Ed25519}}(t'), \ \mathsf{Adv}^{\textit{euf-cma}}_{\text{ML-DSA}}(t'')\Big),$$

with $t', t'' \approx t$. Equivalently, Σ_{hyb} is at least as secure as the stronger *of its two components.*

Proof. Suppose $\mathcal{A}$ outputs a forgery $(m^\star, \sigma^\star)$ on a fresh message with $\mathsf{Vrf}_{pk}(m^\star, \sigma^\star) = 1$. By definition of the conjunction, $\sigma^\star_{\mathrm{ed}}$ is a valid Ed25519 signature on $m^\star$ *and* $\sigma^\star_{\mathrm{ml}}$ is a valid ML-DSA signature on $m^\star$. Build reduction $\mathcal{B}_1$ against Ed25519: embed the Ed25519 challenge key as pk_{ed}, sample pk_{ml} honestly, answer each signing query by relaying the Ed25519 oracle and signing the ML-DSA half locally; $\mathcal{A}$'s forgery yields $\sigma^\star_{\mathrm{ed}}$, an Ed25519 forgery. Symmetrically $\mathcal{B}_2$ against ML-DSA. Hence $\mathsf{Adv}_{\Sigma_{\mathrm{hyb}}}(\mathcal{A}) \le \mathsf{Adv}_{\mathrm{Ed}}(t')$ and $\le \mathsf{Adv}_{\mathrm{ML}}(t'')$, giving the minimum. □

Corollary 10 (Graceful post-quantum degradation)**.** *Let $\mathcal{Q}$ be a quantum adversary for whom $\mathsf{Adv}_{\text{Ed25519}}(\mathcal{Q})$ is non-negligible (Shor's algorithm solves the elliptic-curve discrete log). Provided ML-DSA-65 remains EUF-CMA against $\mathcal{Q}$, $\mathsf{Adv}_{\Sigma_{\mathrm{hyb}}}(\mathcal{Q})$ remains negligible. Thus revealing pk_{ed} on first spend — and reusing the account thereafter — does not enable forgery even under a future quantum break of the classical half.*

Transport. Peer sessions use a Noise handshake with a hybrid static key exchange combining X25519 and ML-KEM-768 (FIPS 203); the shared secret is the concatenation-derived key of both, so recovering it requires solving both the Diffie–Hellman and the module-LWE instance. The handshake is bound to the chain identity $(\mathsf{chainId}, \mathsf{H}_{\mathrm{genesis}})$, so a peer on a different genesis cannot complete it.

7 Genesis Immutability

Let $g^\star$ be the frozen mainnet genesis hash, compiled into the client as a constant (the "binary pin"), and additionally pinned in the chain specification.

Theorem 11 (Computational immutability of genesis)**.** *A conforming node accepts a chain $C = (b_0, b_1, \ldots, b_n)$ only if* (i) $\mathsf{H}(b_0) = g^\star$, (ii) $\mathsf{prev}(b_i) = \mathsf{H}(b_{i-1})$ *for all $i \ge 1$, and* (iii) *each seal meets target. Consequently:*

(a) Presenting any alternate genesis $g' \neq g^\star$ is rejected immediately at the binary-pin check, independent of how much work C carries.

(b) Altering b_0 while preserving $\mathsf{H}(b_0) = g^\star$ requires a BLAKE3 collision.

(c) Altering b_0 and re-deriving a consistent g' additionally requires recomputing valid proof-of-work for all n successors, and still fails (a) at every node, and (c) at the network layer, since peers bind the handshake to $g^\star$.

Proof. Immediate from the acceptance predicate. (a) The binary pin compares the built genesis hash to the constant before any work is considered; a mismatch aborts the boot. (b) By collision resistance of H, no $b_0' \neq b_0$ shares its hash except with negligible probability. (c) The prev-hash chain (ii) forces every successor to be re-sealed under a changed b_0; even a successful re-seal is rejected by (a) at each honest node and cannot peer, by the genesis-bound handshake of §7. The identity is thus enforced by four independent mechanisms — binary constant, spec pin, hash-chain linkage, and P2P binding — plus a release process that refuses to ship a build whose genesis is not byte-for-byte $g^\star$. □

8 Shielded Value (Orchard)

Alongside the transparent account ledger, Sovereign embeds the Zcash *Orchard* protocol (Halo2 proofs over the Pallas curve, no trusted setup). Value in the pool is represented as *notes*; a spend proves, in zero knowledge, membership of a note commitment in an incremental Merkle tree and reveals a *nullifier* that prevents double-spending, while creating fresh output notes (including change). The bundle's signed *value balance* is exactly the transparent value entering or leaving the pool, which is what makes Theorem 7 hold across the shielded boundary. This is a UTXO-like note model layered on the account base — the "outputs and change" world — supplying unlinkability that account reuse cannot.

Remark 12 (Honest scope). Orchard's confidentiality rests on the discrete-log hardness of Pallas, which is *not* post-quantum. Shielded *amounts and linkage* are therefore subject to harvest-now-decrypt-later by a future quantum adversary. Crucially, this is a privacy limitation, not a solvency one: value conservation (Theorem 7) does not depend on the shielded proof system's quantum hardness, so *funds* remain accounted for regardless. We state this rather than obscure it.

9 xusD: an Over-Collateralized Stablecoin

xusD is a DAI-style collateralized-debt-position (CDP) synthetic dollar, native to the protocol. A vault locks collateral c (in xus) and mints debt d (in xusD) subject to a minimum collateralization ratio $\rho > 1$ (initially $\rho = 1.5$) against an oracle price π (USD per xus, in 10^8 fixed point).

Proposition 13 (Vault and system solvency)**.** *A vault is healthy iff $c\pi \geq \rho\, d$. If every vault is kept healthy (by minting limits and, when breached, liquidation), then aggregate collateral value backs all outstanding xusD with a margin:*

$$\sum_i c_i \pi \;\geq\; \rho \sum_i d_i \;>\; \sum_i d_i,$$

i.e. the dollar value of collateral strictly exceeds xusD in circulation.

Proof. Summing the per-vault health inequality $c_i\pi \geq \rho d_i$ over all vaults and using $\rho > 1$ gives the stated chain of inequalities. □

Remark 14 (The honest hard part). Proposition 13 is conditional on two things the mathematics cannot supply by itself: an accurate oracle π, and liquidations that actually execute during

stress. A stablecoin is only as solid as its ability to *sell* collateral in a crash; with a young, thin collateral market this is the binding risk (the "Black Thursday" failure mode observed on early DAI). Sovereign therefore treats liquidations and a surplus backstop as first-class, and treats a live XUS/USD market as a precondition for calling the peg "hard" rather than nominal.

10 Network and Assurance

Peers discover one another by seed and gossip, authenticate with the hybrid handshake, and resist eclipse via decay-based per-IP scoring and connection caps. Every release passes a *genesis double-lock* gate — the pinned hash must be unchanged *and* genesis must rebuild byte-for-byte to it — plus formatting, linting, the full test suite, cross-implementation known-answer vectors, a wallet-key randomness battery, reproducible bit-for-bit builds, and dependency-advisory audits. A standalone adversarial harness throws timewarp, header-tampering, coinbase-theft, forgery, hybrid-conjunction, replay, and equal-work attacks at the real consensus code and asserts each defense holds.

11 What This System Is Not

An unequivocal paper states its limits as clearly as its claims.

- **Not FIPS-140 validated.** Sovereign *implements* the FIPS 203 and FIPS 204 *algorithms* via audited libraries. It has not undergone CMVP module validation; "implements FIPS 203/204" is not "FIPS 140 certified," and we do not claim the latter.
- **Shielded pool is not post-quantum** (§8). Signing and transport are hybrid PQ; Orchard privacy is not.
- **Mainnet is live and not third-party audited.** The current assessment of record is an internal audit; holding value is at the holder's risk until an external audit is completed.
- **Proof-of-work honest-majority assumption.** Theorem 3 assumes $q < 1/2$; a young network has less absolute hashpower than a mature one, which is a real, size-dependent risk.
- **Stablecoin risk is collateral risk.** Proposition 13 is conditional on the oracle and on liquidations executing under stress.

12 Conclusion

Sovereign is an attempt to build money the way a skeptic would want it built: mined by anyone, owned by no founder, signed by a scheme that a quantum computer alone cannot forge, anchored to a genesis that the software itself refuses to alter, and described in a document that tells you where the edges are. The theorems above are the claims we are willing to defend; the disclosures are the claims we are not yet willing to make. Both are the point.

Genesis `cb0272ff88e64c18cde0257f7fae1c8236b02651f10cc7a02456fd682ee2e72d` *State root* `53852c74...6ce6aa`
Source: https://github.com/cloudzombie/sov